THE WORLD'S ONLY DEDICATED GAME MODDING MAGAZINE FREE WITH PCFORMAT 164

gamemaker

create a mod » design new levels, textures & skins » extend the life of your games

GROUND CONTROL II

STRATEGY AT YOUR FINGERTIPS

TAKE COMPLETE CONTROL OF THIS GROUND-BREAKING LEVEL EDITOR

ON THE DISC
ALL THE TOOLS, MAPS & MODS YOU NEED

HOMEWORLD 2

MAKE YOUR OWN GALAXY FOR UNIVERSAL CONFLICT!

SIMS 2 BODY SHOP

CREATE UNIQUE CHARACTERS FOR THE NEXT GEN SIMS

CALL OF DUTY

DESIGN A CORNER OF SOME FOREIGN FIELD FOR THIS TOP WWII SHOOTER

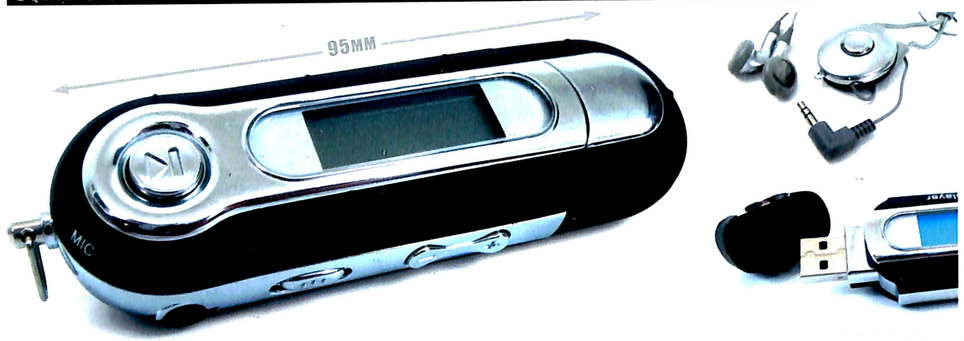
ISSUE 01 » AUGUST 2004

future publishing
Media with passion

PLUS TEXT ADVENTURES BATTLEFIELD 1942 HEAT OF BATTLE TALE IN THE DESERT

MP3 PLAYER / USB PENDRIVE & BUILT-IN VOICE RECORDER

SUMVISION 128MB OR 256MB MP3 PLAYER. CAN ALSO STORE/TRANSFER ANY OTHER FILES



- 128MB OR 256MB MP3 PLAYER
- PLUG & PLAY USB CONNECTION
- SLEEK BLACK & SILVER DESIGN
- SIMPLE NAVIGATION CONTROLS
- 5 PRESET EQUALISERS (ROCK/POP ETC)
- SUPPORTS FOLDERS & ID3 TAGS
- USE FOR DATA STORAGE
- 4.5 HOURS VOICE RECORDING WITH BUILT-IN MICROPHONE
- BLUE BACKLIGHT DISPLAY
- 15 HOURS BATTERY LIFE ON 1x AAA BATTERY (INCLUDED)
- COMPLETE WITH USB EXTENSION LEAD, EARPHONES & NECK STRAP

SUMVISION USB MP3 PLAYER 128MB OR 256MB

128MB £38.99 | 256MB £53.99

INCLUDES VAT - SAVE £36 OFF RRP

MPEG-4 CAMCORDER / 4.1 MEGAPIXEL CAMERA

MUSTEK DV5000 MPEG-4 DIGITAL CAMCORDER & 4.1 MEGAPIXEL CAMERA WITH FLASH!

MUSTEK DV5000 MPEG-4 DIGITAL CAMCORDER

£96.99

INCLUDES VAT - SAVE £33 OFF RRP

- DIGITAL CAMCORDER USING MPEG-4 / DIVX COMPRESSION FOR HIGH QUALITY VIDEO
- RECORDS UP TO 52 MINS WITH ADDITIONAL SD OR MMC CARD
- TAKES UP TO 495 PHOTOS ON 32MB BUILT-IN MEMORY
- 4.1 MEGAPIXEL HARDWARE CCD
- BUILT-IN FLASH & MIC FOR VOICE RECORDING & MOVIE AUDIO
- 4x DIGITAL ZOOM
- 1.5" TFT COLOUR SCREEN
- COMPLETE WITH TRIPOD, BAG, USB LEAD & BATTERIES

UKGADGETS.COM

0870 725 9999

Special Reserve
Discount Network
Specialreserve.com

MAIL ORDER SALES - CALL:
0870 725 9999
SHOP WITH US ONLINE AT:
UKGAMES.COM
OR VISIT ONE OUT OUR AMAZING
8 BIG DISCOUNT SHOPS
OPEN UNTIL LATE

BASILDON
Festival Leisure Park, Basildon, Essex
☎ 01268 287 776

BRISTOL
351 Gloucester Road (A38), Bristol
☎ 0117 924 5000

CHELMSFORD
43 Broomfield Road, Chelmsford
☎ 01245 348 777

EGHAM
168 High Street, Egham, Surrey
☎ 01784 473 444

NORWICH
10/14 Plumstead Road East, Norwich
☎ 01603 300 004

NOTTINGHAM
164 Derby Road, Stapleford, Nottingham
☎ 0115 949 1000

SAWBRIDGEWORTH
The Maltings, Station Road, Sawbridgeworth, Herts
☎ 01279 322 399

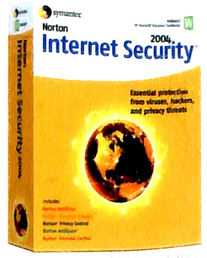
UPMINSTER
Just 6mins from J29, M25. 209/213 St Mary's Lane, Upminster, Essex
☎ 01708 22 55 44

>> STOP PRESS <<

NORTON INTERNET SECURITY 2004

£23.99

SAVE £16



AVAILABLE AT

UKPC.COM
0870 725 9999

>> STOP PRESS <<

gamemaker

gamemaker@futurenet.co.uk
Future Publishing Ltd
30 Monmouth Street
Bath BA1 2BW
Tel: 01225 442244 Fax: 01225 732295
Subscribe: +44 (0)870 444 8659

Editor editor Dan Griliopoulos
Art n' soul of t' party Andy McGregor
Production Ed Libby Pagett

Disc guys! Trevor Witt, Ali Jennings, Jeremy Ford
Software Copyright Co-ordinator Sarah Williams
Contributors Owain Bennallack, Richard Cobbett, Kieron Gillen, Mark Holtham, Neil Rennison, Eve Power, Paul Dean, Nick Mohr, Dan Griliopoulos

Advertising
Senior Ad Manager Debbie Blackman

Marketing
Marketing Manager Fiona Tulley
Overseas Licensing Simon Wear 01225 822798

Production
Production Manager Claire Tovey
Production Co-ordinator Diane Ross
Commercial Print Buyer Mike Thorne

Digital Division
Publisher Kelley Corten
Managing Editor Nick Merritt
Group Art Editor Paul McIntyre
Publishing Director John Weir

Subscriptions & back issues
Tel: +44 (0) 870 444 8659

Customer services
Tel: +44 (0) 870 444 8470
Email: customerservice@futurenet.co.uk

Gamemaker may not be reproduced without our written permission. We cannot accept liability for mistakes or misprints. Letters are assumed to be for publication. We reserve the right to edit letters for space or clarity. Gamemaker recognises all copyrights. Where possible we have acknowledged the copyright holder. Contact us if we haven't credited your copyright – we will correct any oversight.

All contributions to Gamemaker are submitted and accepted on the basis of a non-exclusive worldwide licence to publish or license others to do so, unless otherwise agreed in advance in writing.

Printed by BGP
Distribution Marketforce Tel 0207 633 3300

All contents Copyright Future Publishing Ltd 2004.

Future Publishing Ltd is part of The Future Network plc. The Future Network produces carefully targeted specialist magazines for people who share a passion. We aim to satisfy that passion by creating titles offering value for money, reliable information, smart buying advice and which are a pleasure to read. Today we publish more than 90 magazines in the UK, US, France and Italy. Over 80 international editions of our magazines are also published in 28 other countries across the world.

The Future Network is a public company quoted on the London Stock Exchange (symbol: FNET)

Non-Executive Chairman Roger Parry
Chief Executive Greg Ingham
Chief Operating Officer & Managing Director, UK Robert Price
Group Finance Director John Bowman
Tel: +44 1225 442244
www.thefuturenetwork.plc.uk

future publishing
Media with passion

lil' ducks

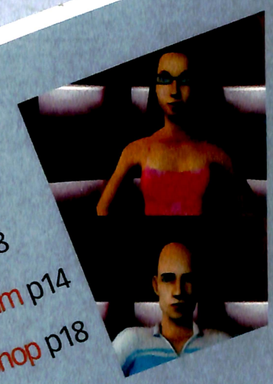
All the twos in this edition of *Gamemaker*, as we explore some of the least derivative sequels the world has supplied. First off, we have the world's first look at the *Ground Control II* editor, *Xed*. Then we've got Eve and Stuart's *Homeworld 2* tutorials, a look at the *Sims 2 Body Shop*, where you can while away a weary week whittling wings for your Sims, and the second part of our *Battlefield: Vietnam* (*BF1942*).

We also have the second part of our text adventure tutorials, the Blender Bites Back, interviews with the guys behind the *Heat of Battle* mod and *A Tale in the Desert*, and the usual awkwardly collated array of news and Odd mods. And no fat ladies.

gamemaker@futurenet.co.uk

Dan

contents



news p4
developer profile p6
ground control II p8
battlefield: vietnam p14
sims 2 : body shop p18
call of duty p20
text adventures p24
homeworld 2 p28
in hindsight p32
odd mods p34

on your dvd

ALL THE ELEMENTS YOU NEED TO MOD YOUR GAMES.

MODS AND MAPS:
UT2004: RED ORCHESTRA
HALF-LIFE: FIREARMS V2.9
OPERATION FLASHPOINT: LEGAWARZ
BF1942: DESERT COMBAT EXTENDED
BF1942: BATTLEFIELD 1918
DUNGEON SIEGE: COPPERHEAD
UT2004: CARBALL
HALF-LIFE: BATTLE GROUNDS
UT2004: COMMUNITY BONUS PACK 2 VOL. 2
BATTLE FOR EARTH

GAME EDITORS:
BATTLEFIELD 1942
CALL OF DUTY
THE SIMS 2
HALO

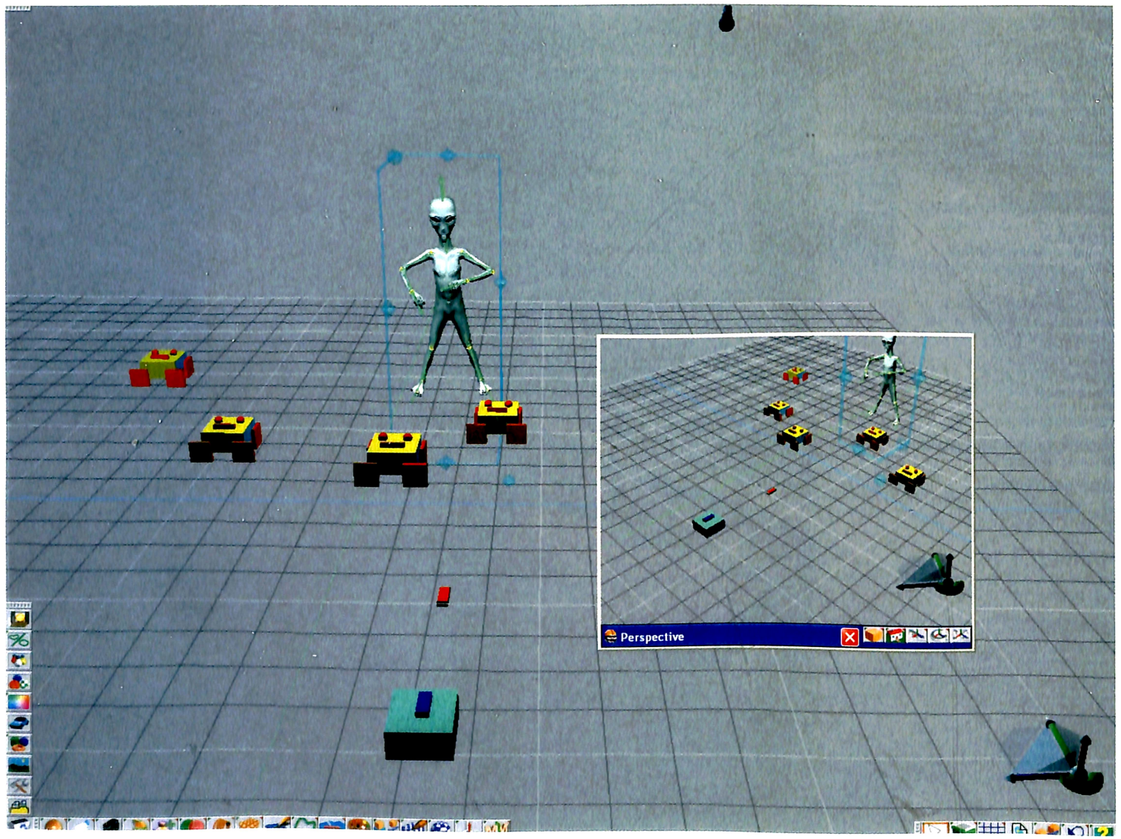
GAME DEMOS:
HITMAN: CONTRACTS
JOINT OPERATIONS: TYPHOON RISING
SOLDNER: SECRET WARS MULTIPLAYER BETA DEMO V2

OTHER DISC CONTENT:
GAMERMAKER TUTORIAL FILES
ONIMUSHA 3 TRAILER
EVE ONLINE CLIENT
BROADBAND NOW
GAME PATCHES

& MORE...

news

HE'S CURRENTLY THE KING OF THE ALPHA RELEASE, WHICH IS WHY NEWSHOUND MARK HOLTHAM CAN'T WAIT FOR THINGS TO GET BETA.



Modeling that 1 million point invader is so easy.

GameSpace Light

It's better to have lopped and pasted, than never to have textured at all.

Modeling is like coding; 3D modeling skills are appreciated in many areas in the computing industry, from hobbyist designs, to game creation, to the latest Hollywood blockbuster. The most commonly known packages run, price-wise, into the thousands of pounds, putting themselves well out of reach of the average Joe. Thankfully most of the developers release limited function freeware versions of the industry standards for us penniless mortals to develop skills with. Joining the likes of Alias's free Maya PLE is Caligari's *GameSpace Light* 3D program. Acting as a cut down version of the relatively cheap *GameSpace 1.5*, *GS Light* has only one restriction, a 650 object polygon limit, scenes up to 5000 polygons and a rendering size of 1024x768. All the other 1.5 features are intact allowing you to reach your modeling Mecca, and all for the grand price of nothing. www.caligari.com/gamespace/

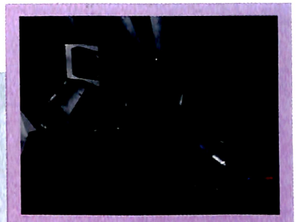
Not what you want to open the door to first thing monday morning.

Cube Engine

Want a fully functioning game engine that offers a little bit more than the average SDK, but doesn't want to pay enormous licensing fees? The gorgeous and powerful Cube Engine may be just what you need. It can handle single and multiplayer environments and is entirely open source and therefore really rather good. Find out more at www.cubeengine.com/

UnWheel

Winning a phase of the Make Something Unreal Contest is no mean feet. Not resting on its laurels, the UnWheel team raced to get their game ported over to Epics latest for a mod showcase at this year's E3. Thankfully the conversion has now been released to the general public for us all to enjoy.
http://unwheel.beyondunreal.com/



Hollow Moon

Gunning in a film noir style game is something of an oddity, yet something that could spawn a unique atmospheric experience, should it be done right. The hollow Moon Team have released a Pre-Alpha of the *Unreal Tournament 2004* single-player mod. Although the enemies are missing, the music and ambient sound effects are already ingame, providing a glimpse at what is to come.
www.blackcatgames.com/

GTA: Liberty

Fed up of waiting for the latest incarnation of *Grand Theft Auto*? Well so are the people behind *Vice City*'s latest addition. Liberty City was the city that players had the pleasure of joy riding around in *GTA III*. The new updated City has taken on the benefits of *Vice City*'s engine, giving you improved eye candy and the ability to fly helicopters.
www.gtaforums.com/index.php?showforum=104

Call of Duty Fusion Pack

Fusion packs are well known and associated with *Half Life*'s Counter Strike mod. This time attention has turned to Infinity Ward's World War II shooter. The package adds new visual flare by the way of new particle and Audio enhancements, high resolution models and new console graphics. The package replaces both single and multiplayer modes.
www.games-fusion.net/

Far Cry SDK

Do your glands flare up and blood begin to boil at the first sight of an FPS game? Well if you haven't yet invested in Crytech's finest, but have the desire to work with the game engine, then your wish may well have just come true. Crytech has announced it is releasing the modding version of *Fry Cry*'s SDK, which is freely available over the net as from 14th June.
www.farcry.ubi.com/index.php

INDUSTRY EYE

Owain Bennallack suggests a brand new way to lift characters out of a book.

After many lonely years, when game developers could no more easily bone up on game development than they could explain what they did to their grandparents, there's now a glut of books available on creating games – from hardcore graphics coding bibles to collections of interviews with leading producers. The biggest difficulty for today's would-be student of making games is perhaps affording such an extensive library in the first place.

Creating Emotion in Games: The Art and Craft of Emotioneering, written by David Freeman, is however unique. Instead of path-finding techniques or clipping planes, Freeman's tome concerns what some think impossible: making gamers laugh and cry. And we're not talking about a vague call-to-arms for a new kind of moving entertainment. Freeman explicitly sets out some 1,500 techniques, divided into 32 categories, that you can use to create emotional attachment. It's like a map of shortcuts to the player's soul.

For example, he outlines the 'Emotionally Complex Situation'. Give the player a princess to rescue by all means, but what if she previously and accidentally killed the hero's brother during a lover's tiff? And with this skeleton in her past, could she even trust her potentially vengeful rescuer? Allowing the princess doubts is what Freeman calls an 'NPC Deepening Technique'.

By layering up the narrative depth and player choice using these and the innumerable other techniques listed, Freeman says you'll create a game that fully connects with the player's imagination: just as watching a film is more emotionally engaging than a sweaty round of ping-pong. You can't forego the action – playing ping-pong is better than talking about it – but you can make relatively simple design decisions to stand out from the crowd.

Is he right? Surely people can't be manipulated by rules? Of course they can't – but the techniques described are more meant to get you delivering real-life style experiences in handy game-sized chunks. We all get emotional every day, sometimes just getting to work. Why should we switch off our feelings when we play?

Emotion-inducing techniques can't replace creativity, even in game development. But it's worth noting that novelists and screenwriters use similar techniques routinely (Freeman originally taught screen-writing in Los Angeles) and creative writing degrees these days are commonplace, boasting semesters packed with this kind of methodology.

And let's face it, creating a giant with an inferiority complex who is scared of heights is more practical than trying to better John Carmack's graphics engine.

www.developmag.com

Artillery pieces are often vital objectives to be held.

HEAT OF BATTLE

developer profile MIKE DOMENIGHINI

THE CURRENT MODS FOR *CALL OF DUTY* ARE UNINSPIRING TO SAY THE LEAST. PAUL DEAN THINKS HE'S FOUND ONE TO GET THE BLOOD PUMPING.

ABOVE Mike Domenighini; the smile of a man who knows.

The multiplayer mod scene for *Call of Duty* is still in its early days. Mod designers have added new levels here, damage tweaks there and sprinkled in a light dusting of new weapons or class-based action but, by and large, the customisable *Quake III* engine behind the game has been underexploited. This makes *Heat of Battle* all the more remarkable; the nine man team behind it are intent on re-wiring the game from the inside out, doing work equivalent to a dozen of the more superficial *CoD* mods.

Despite it being relatively early days for the mod (at the time of going to press we were playing v0.32) it's already looking impressive. Several frankly fantastic new maps have been built specifically for *HoB* – you may recognise one tweaked version of a *CoD* favourite – and each of the four in-game nationalities already has one new weapon to play with. The core capture-and-hold dynamic of the mod is firmly in place, along with wave respawns, rather than individual ones, and a host of minor improvements.

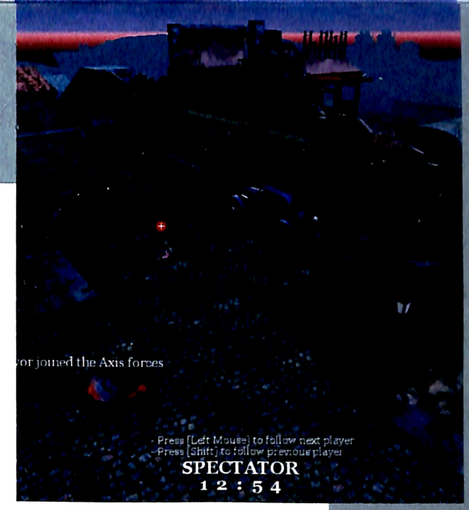
Many mod makers would already be satisfied with the work done on *HoB*, but Mike Domenighini knows there's still a lot to aim for: built-in voice comms system, portable machineguns and bullets which penetrate thin walls and surfaces, to name but a few.

If the *HoB* team can achieve half of their aims they will have accomplished more than any other *CoD* multiplayer mod. Mike is full of praise for the *HoB* team: I have the easiest job in that I get to work with this amazing group of individuals every day…the team handles itself so well. They are amazingly talented individuals.

GM Can you tell us about the team behind *Heat of Battle*?
Mike Each member of the team is required to actively engage in the community. In some cases, the ability to interact with the community carries more weight than the particular developing talent. Matt Blessing is our 2D and 3D artist; he does the skins and artwork for the game. Kerry Dorsey makes sure all of our work can be moulded to help out server administrators as well as those running in the Linux environment, he also maintains our private and public development servers. Jim McDermott has been coding for us since v0.20. Peter McMullan is solely responsible for all the new weapons that our gamers get to interact with – in many instances Pete puts retail models to shame. Adam Gascoine and Jason Marcotte are the team's primary mappers and level designers. Gabe Volker is our jack-of-all-trades: at any given time he's working on improving the code, adding GUI features for server administrators and working on a map. Finally, our historical accuracy comes from Mike Pulaski, who is also the 'Mike' in Mike's Marauders. Mike is a very generous host and the folks at Mike's Marauders have been the catalyst for this project's growth.

GM How much of *CoD*'s content have you modified and why?
Mike Matt has reskinned almost every weapon for the German and U.S. soldiers as well as their uniforms. Our sound effects, while not all recorded on our own, have replaced many of the originals. Also, we worked hard to tweak the weapon values and player speeds. For example, in *CoD* you could carry a Browning Automatic Rifle and move at the slowest pace in the game but, if you pulled out your pistol you would speed up. We blended the speeds down so there's an incentive to move as a unit, and worked to reduce the speed differential between pistols and primary weapons.

GM What are the most time consuming elements of the project – adding new content or modifying initial content?
Mike All of the things that should have been pretty simple caused us problems. I think new weapon implementation took us about a month and a half to get to a functional state and getting the first maps completed with *CoDRadiant* was tricky. Every new mapper to the team probably lost about two semi-completed maps each to



FAR LEFT The Americans get to play with the new M3 Greasegun: more accurate than the Tommy gun.

LEFT The flag indicators at the top of the screen show who holds which objectives. Total domination equals victory.

BELOW LEFT Chilly Russian winters benefit no end from the heat of battle.

BELOW FAR LEFT It always looks so damn easy on paper.

bugs within that software. The most integral components like the gametype code went smoothly in comparison. That could have been because of the talent of our coders but I think the folks who were working on adding additional content suffered from the lack of responsiveness from *CoD*'s developers at the start. Additions like voice comms and bullet penetration require direct support from Infinity Ward. These features haven't been time consuming because we can't even step into development without more resources.

GM What tools do you use to add content and modifications?
Mike Pete uses *3DSMax* to develop his highly detailed models, and turns that work over to Matt who, with Pete's Meshes/UV Maps, creates original skins for those models with *Photoshop*. Our mappers utilize *CoDRadiant*, while our coders will use anything from open-source coding apps to *Notepad*. We use an SCP client and server as our mutual storage space and that's made the biggest difference in overall productivity. Before, we were swapping files back and forth over MSN, personal FTP's and e-mail, which proved to be very damaging to our work so, with the help of *WinSCP*, we've been able to work as if we were in the same office.

GM Do you think you can achieve all your aims? Have you had to abandon any of your ideas so far?
Mike Unfortunately, a lot of our progress has been hampered by a general lack of 'mod-ability' inherent in *Call of Duty*. It's fine if you want to spruce up weapons, change a few game rules and alter the environment a little. If you really want to change the way the game is played or add your own dynamic elements, you're a bit strapped. For example, we always wanted to add a highly functional control panel for server administrators. This would allow an admin to simply select the player name they wish to interact with and then click on the command they wish to use. What we've learned is that there is no clear way to interact with these menus and we've had little luck getting in touch with the original developers in order to determine feasibility. We've found a way to implement a majority of our desired features. Unfortunately, few of these methods of implementation stand up to the level of quality we'd like to provide.

GM What are the greatest challenges you've faced so far?
Mike Balancing 'real life' and the development project. This becomes most apparent around release time; you start to see how the time spent working on the game takes its toll on your free time. The other real challenge was working with the original developers of *CoD*, it's hard to contact them due to their workload; the lack of communication came as a surprise to us all.

The easiest part has been working with our community. Our gamers are probably the best in gaming; I visit the forums of other retail and modified games and it can be a cluster of arguments, demands and vulgarity. This is what I'm most proud of: if you roll through our forums you see a group of people offering constructive criticism and being a part of the development team. Finally, I really have to hand it to GameSpy for the work they've put into helping us out: they made the promotional work a breeze.

GM Where do you see modding headed in the future?
Mike In the past the real division between modders and professional developers has been that professional developers have thousands of dollars worth of software to work with while modders generally used whatever they could get their hands on. It's a very exciting time for modders because we're seeing the big development companies providing professional tools geared towards their environment. The next generation of games and mod development tools is going to take that newly found quantity and add a quality that I don't think any of us could anticipate.

THE SINCEREST FORM OF FLATTERY

Day of Defeat V2.0?

***Heat of Battle* may not be entirely original in its approach, but its building a new spirit of co-operation within the *Call of Duty* community. Mike explains: Matt Blessing and I had been playing with Mikes Marauders and Mike's always leaned towards WW2 shooters that focused on teamwork. We had all played *Day of Defeat* but Steam left many of our gamers out in the cold and we wanted to reproduce that game's co-operative nature in *Call of Duty*.**

This meant working towards a proven system of co-operation, the positives of battle; brotherhood, teamwork and an objective-oriented mindset. Playing any WWII game without these seemed unnatural. As much as we trumpet the individuality of *HoB*, we'd be lying if we said we weren't heavily influenced by *Day of Defeat*.

Mikes Marauders site is at www.mikesmarauders.com. Their Heat of Battle server is at 209.43.38.121:28960.

Ground Control II

TAKE CONTROL OF *GCII: OPERATION EXODUS*'S POWERFUL 3D MAPPING TOOL XED WITH OUR EXCLUSIVE TUTORIAL. COMPILED WITH THE ASSISTANCE OF THE FOLK AT MASSIVE.

Ground Control II itself may seem merely solid, but with the addition of its mission editing package, *Xed*, we can make it truly exciting. We were privileged to be the first to get hands-on with *Xed*, in the offices of Massive Entertainment in Malmo, Sweden, where we were taken through the paces by none other than Per Edman, lead mission designer on both *Ground Control I* and *GCII*. Per has many years experience in the industry, and one fine day in Malmo, he took it upon himself to guide us through the steps needed to get a multiplayer skirmish map up and skedaddling in *Ground Control II*.

There are some key things to be noted about *Xed*; just like *Ground Control II*, it is very processor intensive at the higher settings: if you're having trouble, just lower all the settings in the Preferences menu. *Xed* also has a predilection for opening too many large unwieldy windows, and not shutting them when you've finished with them. This is a piece of editing technology that really benefits from a multi-monitor set-up, even if that's even more of a drag on your already overworked processor. Finally, don't expect your first test map to work initially; there are more variables involved than the British economy, and you don't have a hundred economists holding your hand to guide you through.

Dan was caught doing no work one time too many in the *PCF* offices, and as punishment was sentenced to ten years hard labour, breaking games and making rocks. He is however an expert in his field, honest.

ON THE DISC

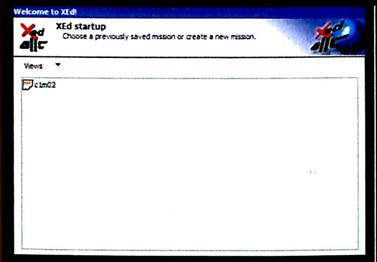
DIFFICULTY INTERMEDIATE

TIME TAKEN 3 HOURS

WHAT YOU NEED...
Xed, bundled with *Ground Control II*

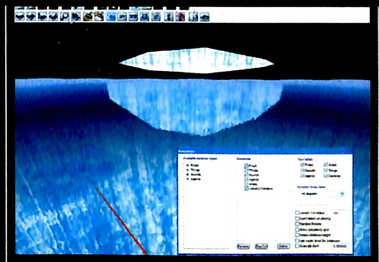
// HIT THE GROUND RUNNING

Making a messy map is easy; making a splatmap isn't much harder.



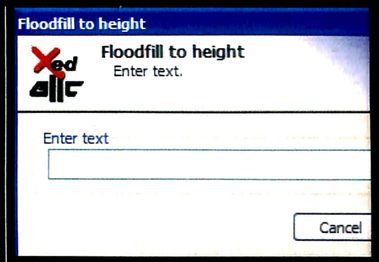
01 IN THE BEGINNING

Load up *Xed*. You'll be confronted by a window asking if you want to open an old map or create a new one. At the bottom of the window, select the locale from the **New mission pane** (Battleground is a cityscape, Farmland is swampy, the rest are fairly self-explanatory.) This establishes the texture set you'll be using for the mission. We recommend you don't select the orange map, as this is solely used for testing height-maps, where it goes into red is impassable to land units.



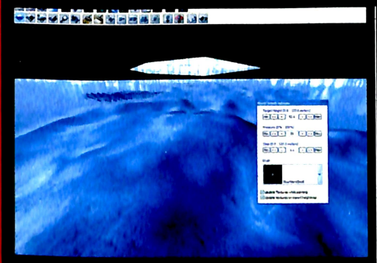
02 WATER LOT OF WATER

Having clicked the new button, you'll be presented with a window full of water, and a strangely familiar, but obviously wrong, sky. It's currently not possible to change your terrain type so the texture combination you see here (the splatmap) is what you'll be stuck with for the rest of the tutorial, start again now if you want a change. To quickly replace the sky, click **Commands ➔Select Environment** and then select a time of day.



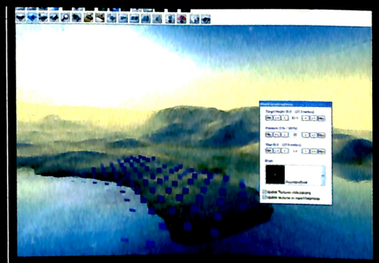
03 LAND AHOY!

Isn't that a lovely sight? First we'll need some land, unless you plan your units to be wading. There are various ways of doing this. The first we'll be exploring is found under **Commands ➔Flood Fill**. This sets the land over the whole map to a set level, not particularly interesting, but very useful for all terrain maps or shallow water maps. The normal water level is around 30, so make sure that you take this into consideration when planning your map.



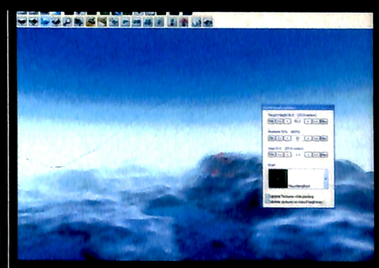
04 ARTISTIC LICENSE

The first useful tool, once you've established your map (see the column to the side for an alternate method) is **WorldPaintToHeight**, the first button on the second row (shortcut Z.) This pressure paints to a set height – you can set the height either by typing into the **Target Height box** of the **World brush options** window. If you select a different shaped brush, like **MountainsSpider**, it'll produce a different effect.



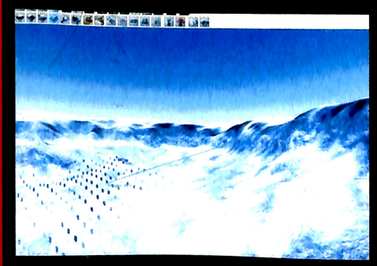
05 AIRBRUSHING HISTORY

The **WorldPlusMinusTool** (shortcut X – the second tool on the second row) is the most useful tool if you're intending to do your whole map by freehand. Painting with this will raise the area depending on the pressure, to a ceiling of 127 units, while holding **[shift]** will lower the area. Just like the previous tool, it's effects are dependent on the brush shape you use (which you can select from the **World brush options** window).



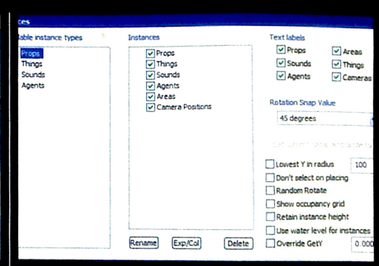
06 SAMPLING

The **WorldSetHeightTool** (shortcut C) is the third button on the second row, and acts like the eye-dropper in a paint program, by left-clicking on the land it can sample the land height immediately. It will sample the height under the centre of the brush under normal circumstances. If you hold down **[shift]** whilst using this brush, it will snap to gridpoints (we'll go into more detail on that later.)



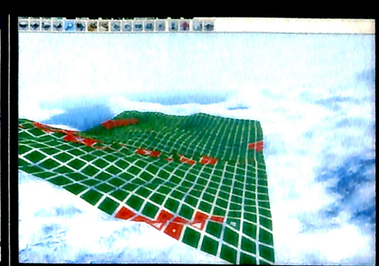
07 RAMPING UP

A tool which experienced terrain mappers will find uniquely alluring is fourth along on the second row: this is the **WorldRampTool** (shortcut R), which, depending on pressure, can create a smooth ramp from one point to another. 100 per cent pressure will create a perfect ramp between the two points, although it also respects the shape of the currently created brush, a fading airbrush will produce a curved ramp, while a straight line or hard edged brush will produce a hard edged flat ramp.



08 WATER RETENTION

If you're already wondering what the **Retain instance** tick box in the **Instances window** means, it means all the **instances** (the items, buildings, trees and so on) that are placed on the map will not move when the terrain beneath them is remodeled. This isn't really necessary until you come to the final tweaking, or rebuilding of the terrain to accommodate bases and the like, but it can make a big difference if you're looking to bury instances for special effects.



09 GET AROUND

Once you have successfully completed modeling your terrain, it would be wise for you to want to check your landscape for manoeuvrability. Using the **Inspect tool** (fifth along on the bottom row), you can soon discover where the land units can go, if you hold down **[shift]**, it will tell you where amphibious units can go, and if you hold **[control]** it will tell you where any infantry goes (for example, including woodland pathing, which infantry can pass through.)

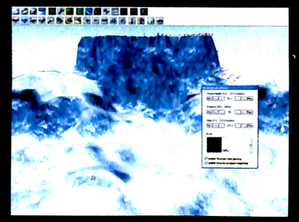
FRACTAL LANDSCAPES
From chaos comes order, or such tosh.

Remember those Mandelbrot chaos generators way back in the 1980s, that sucked up power just so you could zoom into an edge far enough to see the same thing again and again? You can create a map by a similar method in *GCII*. Look at Commands ➡ Generate Fractal Heightmap for the menu screen. You can see this window has a somewhat-daunting six tabs. Don't worry, you don't have to use all six of them at once; they're best used on different texture sets, as some of them produce more flowing environments, others produce jagged or canyon-riddled textures (with the splatmap automatically updating to the terrain structure underneath it.)

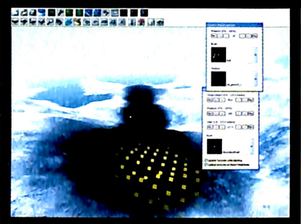
THE PLAYFIELD
Thinking inside the (big red) box.

Admittedly, we're all simple players, strutting our stuff on the stage, but we still need a stage. These first two pages have shown us how to set up the scenery effectively and quickly. The red box you see on the 2D Map [Shortcut shift + F6] is the playfield; within which the player can move in-game. You're not so limited in the editor mode and you can get right to the edge where the map plunges back into the endless sea. To alter the playfield co-ordinates, click the fifth button along on the first row [shortcut 3]. The co-ordinates are listed in terms of distance from the origin (which, perversely, is in the top-right of the 2D Map.) If you want to make maps of curious shapes, or lengthen dropship arrival times, this is a quick way to do so.

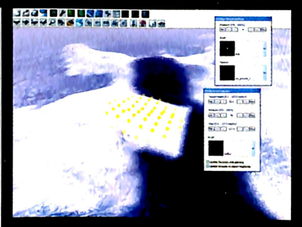
// A TEXTURE COOKBOOK
Making your landscape look ruggedly individual is easy as pie. Mmm, pie.



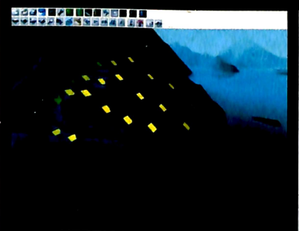
10 ROBO-TEXTURE!
First select the **Texture Paint Tool** (shortcut V). Go to **View ➡ Texture Brush Options** (shortcut **[Shift + F3]**) so you can see the palette you'll be painting with. You'll see you also have a second brush box, in addition to the brush shape box, to paint textures. When you paint a heightfield, the texture will change automatically to represent slope and altitude; with texture painting you can update those textures to more customized ones.



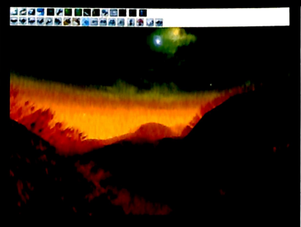
11 TEXTURE SETS
Depending on the terrain types, there will be four to six different textures available. Just like the other painting tools, there's also a pressure setting allowing you to treat the tool like an airbrush; it's best to start with low pressure: Per himself starts with 3 to 10 for finessing, although other level-designers start higher. Even these textures can look tiled as they have repeating patterns, but careful blending and retouching can help.



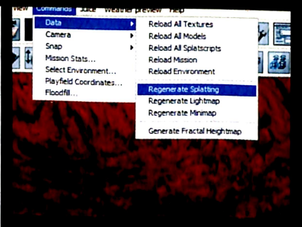
12 TEXTURE ERASING
This, the second texture tool, allows you to revert back to the standard splatmap default texture: handy when you've made a serious error when painting your textures. Once you've retextured the map as you like it, deselect **Update Textures** while painting in the **World Brush Options** window, along with **Update Textures** on import heightmap. This means further edits to the heightfield won't affect the texture painting you've done.



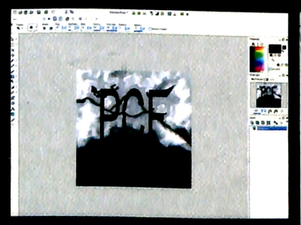
13 VERTICAL TEXTURES
It's important to note that the bottom two textures in the **Texture pull-down box** in the **Texture Brush Options** window are sometimes intended for vertical surfaces only. One is intended for East to West facings, the other North to South facings. The flatter map texture sets (such as farmland) don't tend to have these facings, so you should try to avoid too fierce height changes with these (or accept that sheer surfaces will all look fairly similar.)



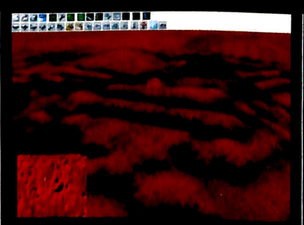
14 AUTOPROPS
Not all textures have these, but you'll soon notice if they do. Autoprops are the strands of grass on most of the farmland textures, or the twisted girders, metal poles and cinder blocks found on the **Battlegrounds texture set**. It's very important to notice that these **autoprops** have no effect on the **battlefield** except a purely aesthetic contribution; they don't block any movement.



15 TALKIN' BOUT REGENERATION
Sometimes you'll have a heightmap that you're enormously proud of, but that you've covered in crude daubing to make it look less like a *GCII* map and more like *Perimeter*'s blancmange maps. Rather than saving out the heightmap, or erasing every element by hand, you can wipe it easily. Click on **Commands ➡ Data ➡ Regenerate Splatting**, and all the splatmaps will be returned to normal.



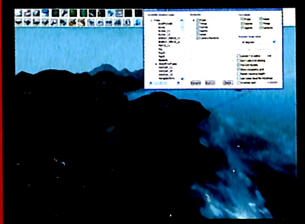
16 CUSTOM HEIGHTMAPS
It doesn't always work just one way; if you want to keep the textures, but feel your heightmap is unrealistic, why not replace that? There's two ways of doing this; paint a height map in a program like *Paint Shop Pro*, or create a map in *Xed* itself and try and export it. There's currently no function to do this, but if you create your map and navigate to the file (it'll be a .tga file in the **map folders** under **custom maps**) you can edit from there.



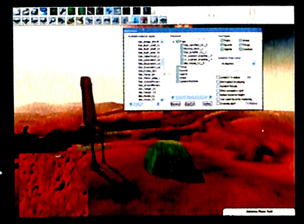
17 RULER
If you're looking to make elements of particular sizes, you'll find that the **Ruler tool** will become a very important part of your work. If you just click and drag the pointer, it'll count the distance between the start and end point. It'll also calculate altitudes if the points you click on are hills. You'll also need to use this when setting the spawn points for the **dropship** later on, as they need to be a set distance from the edge of the movable map.

18 WEATHER OR NOT
This is one of the simplest effects to institute: click on the **Weather preview ➡ cycle Weathers button**. This will move you through the various weathers available for that map, altering sky visibility, and the various effects across the map including apparent view distance (the real view distance stays much the same.) Further, certain weathers add effects like rain and gusts of sands. The actual weather effects can be set later.

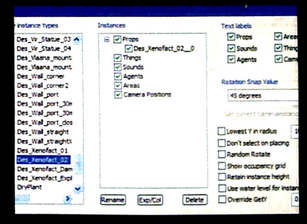
// INSTANT INSTANCES
You had a plain map, just add water (and some instances) to get further.



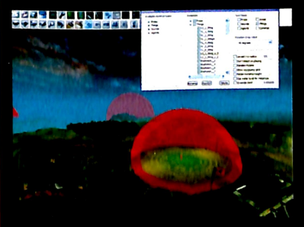
19 PLACING INSTANCES
Let's place our first instance; select the **InstancePlaceTool** (shortcut Q, fourth from the right on the bottom row.) To place it, simply click on the name in the **Instances pane**, and click on the landscape – it should appear there. If you want to rotate it, hold control before you place it and rotate the mouse. The first type of object we'll place is a prop. These are inert objects with collision that occupy space. Included in this category are buildings, trees, vegetation, rocks, and walls.



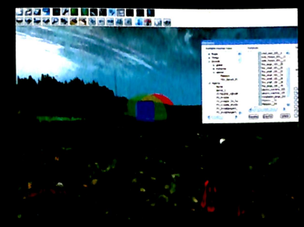
20 PROPER TEA
Since props can be a building, they also have an entering property sometimes. In addition, props have an occupancy grid (you can see this when editing instances by clicking the **Show occupancy grid** tick box on the **Instances pane**.) White doesn't block anything; infantry can pass through black; and nothing can pass through red. So, for example, gate props would need to have red posts (nothing can pass through) but white centres (doesn't block anything).



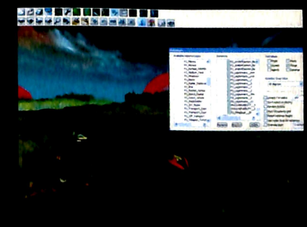
21 IS THEFT
To access the instance information pane, simply click the **InstanceEditTool** (shortcut F). This tool is the major one you'll need to access: it allows you to select, rotate and move instances. To select an **instance**, simply **click** on it – to add more instances to the current selection either **shift-click** them or drag-box them. To move instances, hold **[Alt]** and drag them; most useful is the height alteration technique: hold **[Shift]** and use the mouse wheel to raise or lower the **instance**.



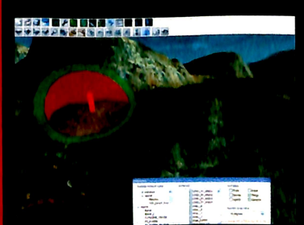
22 THINGS
Nothing John Carpenter here; **things** are those instances which cannot block, have no occupancy grid and no height. The classic example of a **thing** is the **Landing Zone marker**; it changes colour when stuff lands on it, blinks and so forth, but is in itself completely other from the world it resides. **Things** can be confusing as they're activated from scripting; a handle has to be added to them, so we'll leave them until the scripting section.



23 SOUND CONTROL
All positional sounds are placed in the world, just like things, and affect the area around them depending on their classification. Global sounds work well placed anywhere, whereas their mechanical counterparts work best when they're placed near bases or machinery. Again, these are automatically added to the script when placed, so don't place these until you've done some basic scripting.

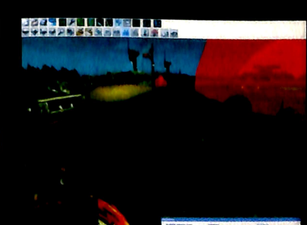


24 AGENTS GRATUIT
Agents are units: anything that has health and can aim or move or both. (Reproduction isn't a criterion for life here). Each has a little chevron marker and is allied with a particular player. They can move in different ways: stationary, perambulatory, infantry, or aircraft. When they're placed, they need to be put into a viable position – you can't put infantry or a vehicle underwater, as the game might crash with an illegal agent.



25 AREAS
Another form of instance, **areas** have a position and radius – the shortcut for the **InstanceAreaTool** is **[Ctrl + Q]**. In order to place an area, you need to click and drag to place a centre point and select the radius. Areas are used to define the size of the victory locations or landing zones, and are used prominently in scripting, when units enter areas or for patrol patterns by the AI.

26 CAMERAS
Placing a camera is simplicity itself: either click the camera button on the top row or press **[F12]**. The direction and position you're seeing at that point will be where the camera points. You can find a camera by opening the **instance pane**, and clicking on its name (it'll be highlighted on the map.) And yes, this really is a true digital camera – no working parts!



27 TOOL UP
A final handy pair of tools when dealing with all these instances, are the **InstanceGrepTool** (shortcut U) and **InstanceCopyTool** (Shortcut Y.) The former selects all objects of the same type on the map, while the latter duplicates them; in combination with the random orientation checkbox: handy to quickly populate an area, without the regularity of the big tree blocks.

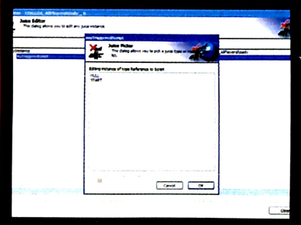
NO UNDO!
Take off your undies. Undoes. Ahem.

Yes, [Ctrl+Z] is in there, but don't instinctively reach for it. The code controlling it is a little laggard with a lot of idiosyncrasies. Sometimes Undo will do exactly what you want it to, other times you'll find various things you did ages before have suddenly vanished, along with everything you've done in between. Manually deleting things is safer than using the dangerous Undo, lest you yourself be undone...

UNRESOLVED REFERENCES
Not all errors are bad errors.

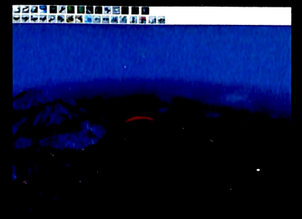
On the next page, we're going to talk about scripting. In the scripting window there's a pane which lists where there are loose scripts that don't point anywhere. When a script is deleted, this window lists all the references, triggers, commands and instances that point to NULL. Double-clicking the it brings up the Juice editor, which will be complaining about the lack of a target, providing a handy shortcut to fix the problem. Note – victory locations have default NULL values because they're cut down versions of LZs – so don't worry about these (unless your map won't load!)

// SCRIPTING MADE SIMPLE
Now that's a straightforward lie if we ever heard one.

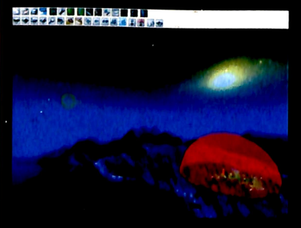


28 THE TRICKY BIT
First, we need to create a script. Open up the **Scripts window** by clicking on the fourth button from the right on the first row (shortcut **[Shift + F5]**.) You always start with a **Start script**. Inside the **Start script** set up one trigger by double-clicking on **TRIGGER_AllPlayersReady**. Set **MyTriggeredScripts** to **Start**, and this should lead to the level beginning when all players are ready for it to start.

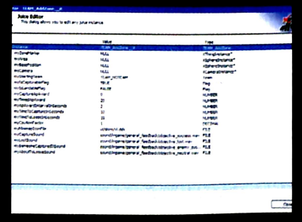
29 ZONE ENDERS
Create a script called **zones** or **Zones_LZ** for the **landing zones**. Select the area tool, and create an area 47 units across for one of your LZs. Rename it **LZ_1_area** (the names are logical, they don't affect the script) Now open the **InstancePlaceTool** and navigate in the **Instance pane** to **Things ⇒ Global Prop Types ⇒VL_thing**. This **thing** is the visible marker for the **LZ** or a **VL** (victory location.) Rename it **LZ_1_thing**.



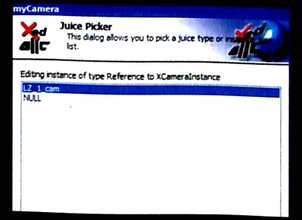
30 DROPPING BY
We also need a starting area for the **dropship**. Go back to the area tool and place a new area where you want the **dropship** to spawn. Name this **LZ_1_base**. This should always be 500m outside the playfield and each of these should preferably be equidistant from its equivalent **LZ_x_area** otherwise certain players will have a distinct advantage over others (though this can be helpful in certain scenarios.)



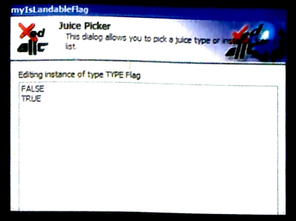
31 WATCHING LZ
You also need a camera at each **LZ** for when the game starts. Place your viewpoint as if you were watching the **LZ** and hit **[F12]** to create a camera. This is the camera you will start with at the beginning of the game and can choose to return to at any time within the game by simply pressing the shortcut W. Rename this camera to **LZ_1_cam**. Now stand on your head and wave your arms. [*Dan, stop abusing your power, that's my job* – Ed]



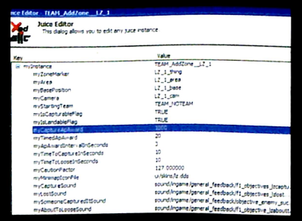
32 THE LZ ITSELF
To create the **Landing Zone**, open the **scripts window**. Select the **Zones tab**, and select the **TEAM_AddZone zone**. This opens the **Juice editor window**, the topmost attribute is **my zone marker**. Double click this to bring up the **Juice picker**. Select **LZ_1_thing**. My area should be set to **LZ_1_area**. **My base position** is where the dropship will idle when not active, and where it'll return. Select **LZ_1_base** for this position.



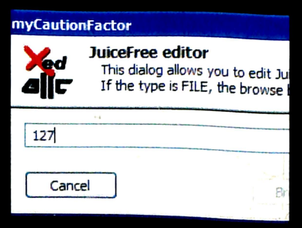
33 JUICE LOOSE
The fourth attribute is **my camera**. When you select this in the **Juice picker** you get a list of all the cameras in the game: pick **LZ_1_cam**, which we just created. The fifth value is **MyStartingTeam**. This determines who owns this zone at the beginning of the game. **Team_NOTEAM** leaves the zone neutral at the start of the game. Change it to **Team_1**. **MyIsCapturableFlag** determines whether the zone ownership can change from team to team (not player, notably.)



34 VICTORY OR LANDING
The next variable is key, **MyIsLandable** flag determines whether dropships can land there, and hence is the only thing that distinguishes a **Landing Zone** from a **Victory location** (apart from its assigned **Thing**.) No zone belongs to a player properly, zones belong to teams (to prevent multiplayer cheating on the larger rewards obtained from capturing zones.) If you want extra victory locations, you can create them in the same way as LZs and change this variable



35 RESOURCE MANAGEMENT
The following five numerical variables are the associated AP values for the zone. The **Capture variable** gives AP every time the zone is captured. The timed income you get while holding the zone is dependent on the next two variables, you will get (first number) APs every (second number) seconds. **TimeToLoosInSeconds** is the initial time taken to make something neutral, **TimeToCaptureInSeconds** is the time it takes on a neutral location to turn it to your side.

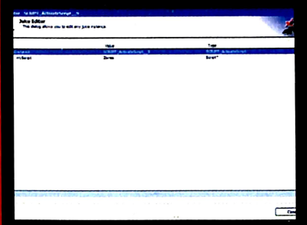


36 MYCAUTIONFACTOR
Although, as we'll see in a moment, the AI hint map can be altered through user intervention, to further incentivise the AI, you can also set the **MyCautionFactor** for VLs and LZs. The value starts at **1** and the higher it climbs, the more the cautious the AI will become in trying to catch it; likewise, the lower it gets, the more reckless it will get, sending small unbalanced groups of enemies to capture it. A default score will leave the judging up to the AI.

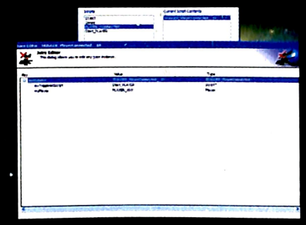
MINIMAP
Not just a pretty picture but a useful tool too.

Although you can bring up a 2D map in the editor by hitting [Shift + F6], the 3D minimap is much more useful; you can bring this up by saving the map, and it should be automatically generated. Not all the icons for victory locations will be displayed on it properly yet. However, this is no problem; when you set up a VL script the icons will be automatically generated. However, Landing Zones will also have the same VL icon, so open up the script for each LZ, and change the MyMinimapIconFile to ui/skins/lz.dds. When your game is eventually loaded up, everything should work fine.

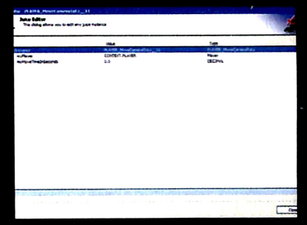
// TYING EVERYTHING TOGETHER
More scripting than a Hollywood bar.



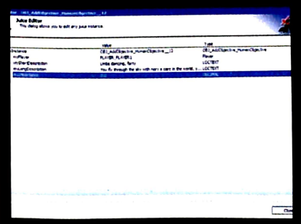
37 STARTING START
Go out and change the name of the add zone to **TEAM_AddZone__LZ_1**. Once you have done this, turn your attention back to the **Start point**. In order to start the zones; inside the **START Script** place a **SCRIPT_ActivateScript** and set it to start the **Script Zones**. Now take a breather for a second… you ready to commence again? Let's continue.



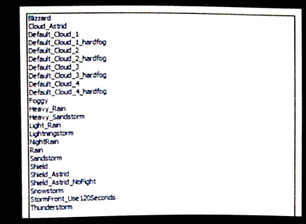
38 GETTING PLAYERS
Create 2 more scripts called **PLAYER_CONNECTED** and **Start_Player**. You always want to create the **Start_PLAYER** script at the the same time. Inside the **PLAYER_CONNECTED** script we have created a single command called **Trigger_PlayerConnected_1**, which points to the second script **Start_PLAYER** and waits for **myPlayer = PLAYER_ANY**. When a player connects it'll start the **Start_PLAYER script**.



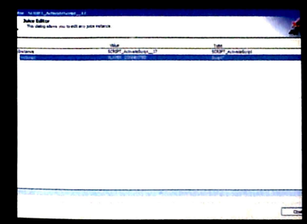
39 LIGHTS, CAMERAS…
Go inside **Start_PLAYER**. This script has been run when any player has connected to the game. We want therefore to move this player's camera to the **LZ**. To do this, we need to find the command **PLAYER_MoveCameraToLz**. The settings for this command determine which player should be moved; set it to **CONTEXT.PLAYER**. Set the move time to **zero**.



40 OBJECTIVE VIEWPOINT
We also want to tell the player what his objectives are. Under **OBJ_AddObjective_HumanObjective**, enter a short and long description for this objective. These will display at different points in the loading process and game; explain exactly what the player is meant to be doing at these points, whether it's simply bludgeoning the opposing Terrans and Virons into submission or something more complicated that you've created through scripting and the mission statistics window.



41 CLEVER WEATHER
Add an **ALL_PLAYERS_READY** script, and trigger it with the start. If you fancy, the **ALL_PLAYERS_READY** script is a good point to change the weather, with the **WEATHER_Set_WeatherEffect** function. Set a default weather type at this point, as it'll start with the game. You might also want to have global soundtracks kicking in, fireworks going off, or any other effect or action that you would like to hear creating an atmosphere as soon as the map starts.



42 GET CONNECTED
Lastly, we need to reactivate the **PLAYER_CONNECTED** script. To do this we add the **SCRIPT_ActivateScript** function to the **START_PLAYER** script and point its only attribute, **MyScript**, towards **PLAYER_CONNECTED**, so the script is restarted for the next player who joins. Now replicate the key parts of this process, creating a couple more landing zones, and some victory locations to squabble over, and your map should be near completion.

IMPORTANT COMMANDS TO REMEMBER

MyName, My Description
Name and description; just what it says – this is the map name and text that will be displayed in the joining Lobby.

MyMissionType, Initial AP
The former should be set to Multiplayer and the latter defaults to 7000, which is a reasonable number.

Score limit and time limit
These should be settable in the lobby, so best to leave them and people will set their own values.

MyWindDirection
The way the wind blows, which affects smoke, waves, and a buncha stuff. This is a vector, so the Y value doesn't matter.

MyPlayers
Under this we find all the eight players available. In single-player, normally the first three are human, but you can ignore this for now.

MyDropshipStartLevels
The defaults for this are zero (which is bad.)

MyAIConfigFile
This is necessary for a multiplayer/skirmish map. There always needs to be AI on map, connected to a valid .ice AI file. Experiment!

Attributes to player
Human, AI, script players with no skirmish AI at all except shooting.

Faction
NSA, Virons, Terran Empire. Last non-player faction. Massive have left this free-text in case you choose to add a mod in later.

Team
Here you can limit the teams of available players to the teams available on the map.

MyBasePosition
A redundant statistic. However, there is currently a bug making the very first AI dropship come from here, so give it a position.

Colour
You could oversee this but no need – it should be auto-assigned on joining a game.

MyAllowNoVLDefeat
Toggles that if you hold all victory locations you win the game.

MyDropshipIsAvailableFlag
Determines whether a player has a dropship or not.

MyDropshipRespawnTime
Determines how long it takes to get a new dropship after it has been destroyed. It defaults to 300 seconds.

MyDisallowedMelds, MyDisallowed Units
These options disable certain builds or melds.

MyDisallowedSupports
Disables certain support fire – smoke, artillery, etc.

MyLoading Screen image and text
Customisable; we'll show you how next month.

Final five
Victory/defeat conditions.

MyAllowLZDefeat
Toggles that if you lose your LZ you lose the game.

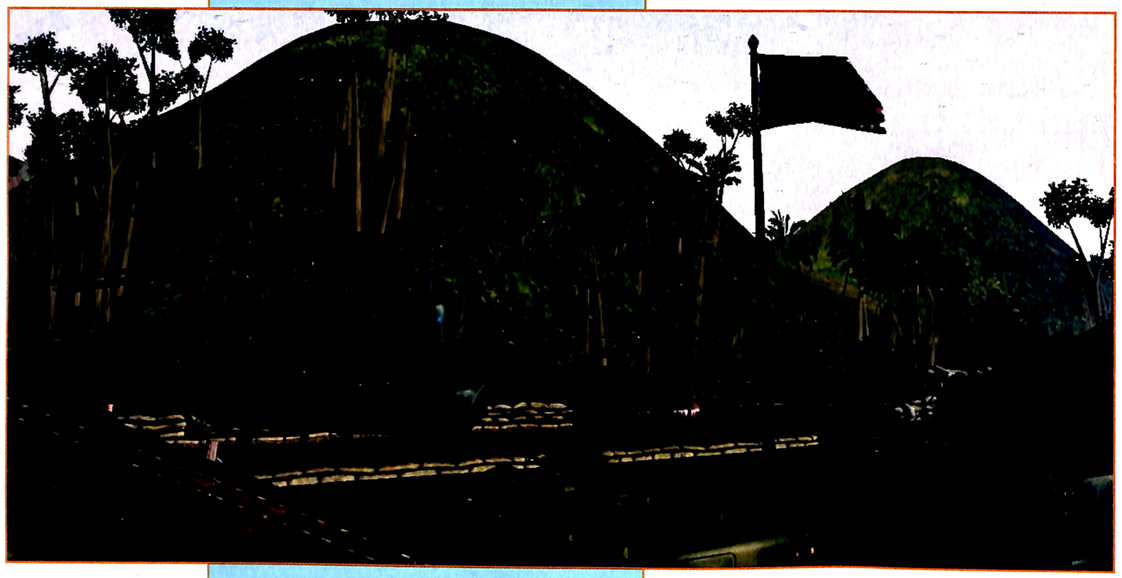
MyDropshipMaxTotalAttributeLevels
Normal maximum of five; can be increased more, but there's no graphical representation on the screen.

MyAllowNoUnitsDefeat
Toggles that if you are the only one with units left on the map, you win.

MyAllowNoPresenceDefeat
Toggles that if you have no units left and no landing zone you lose.

STATS CANIS
Check out the mission stats for a flying finish.

If you take a close look at the box at the bottom of this page, you'll see a bundle of unintelligible commands. These are mostly to be found inside the Mission Stats window (Commands ⇒Stats). They control certain elements of the game allowing you to change the set-up of most of the important background variables. Once you've set these up to your satisfaction, you'll need to export (Shift + X) the map and save it. Now go into *GCII*, and try and load it up. We'll be dealing with troubleshooting and AI mapping next month, as well as how to make a custom loading screen.



ADVANCED TECHNIQUES IN BATTLECRAFT

PART TWO

BATTLECRAFT, IN BOTH THE *1942* AND *VIETNAM* VERSIONS, CAN CREATE REALISTIC EFFECTS WHEN USED CORRECTLY.

BattleCraft, the official editor for the popular multiplayer online games *Battlefield 1942* and *Battlefield Vietnam*, has all the tools you need to create your own custom maps. It may not, at first, be obvious how to use them, but once you see how powerful and easy to use they are, you'll witness their benefits and endless possibilities. Although no automatic process is perfect, with a little adjustment and tweaking, the amount of time saved can be invaluable. The road tool, the surface map painter, and the prefab manager can save an enormous amount of time; time that you can then spend texturing, placing objects and balancing the map to make it all the better.

Loren G. Dobert has been modding games since *Wolfenstein 3D* in 1994. Working with the *BattleCraft42* editor for *Battlefield 1942*, his main source of experience is examining the original files of the game, plus visits to several BF1942 editing forums.

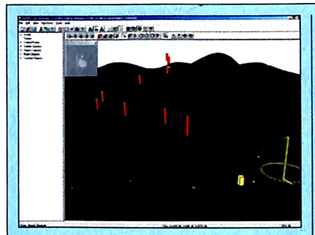
DIFFICULTY INTERMEDIATE

TIME TAKEN 1 HOUR

WHAT YOU NEED...
BattleCraft42 or *BattleCraft Vietnam*. *Windows paint* or other image editor.

// CREATING ROADS, EDITING MATERIAL, AND FINALISING TEXTURES

It's a long journey, made easier by using the right tools along the way.

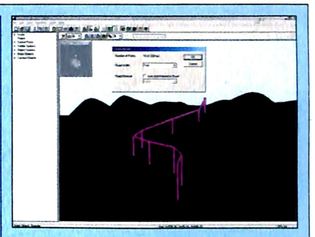


01 PLANNING THE ROAD

Once you have your general terrain editing done, and want to add a road, it's best to determine its start and end points beforehand. Place an object where you want each major turn to be, so you can have a guide to go by when placing road markers. In the screen shot, you can see the jeep as the starting point, and the flagpoles mark where the road will go. You don't have to do it this way, but a guide will be helpful.

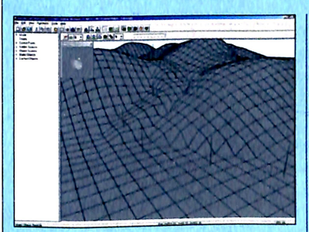
02 STARTING THE ROAD

Switch to the **Terrain Mapper** and click on the road tool button. Hold the control key down and click where you want the first point of the road to be. Click again at each point you want the road to turn. Remember, the road tool can only place a point at the centre of a grid square. Once you are done, release the control key. If you click in the wrong spot, just release the control key, cancel the dialog and start over.



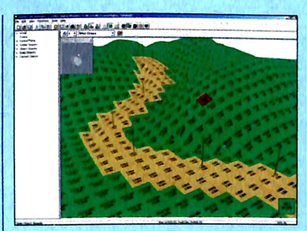
03 THE ROAD PROPERTIES

Once you have the road where you want it, release the control key and you'll get the **Create Road dialog**. Decide how wide you want the road and pick the choice from the drop down list. You can also add the material automatically, which will be helpful in tweaking the material later. You can also use the road tool as a type of narrow terrain flattener, in which case you won't need to fill in the material.



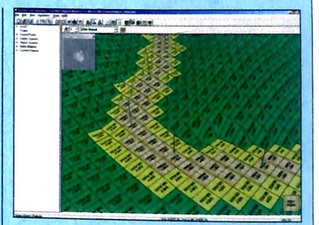
04 EDITING THE TERRAIN

Once you hit the **OK button**, the road will be created, and the material added. This is where you need to adjust what the tool has done to better fit the look you're trying to achieve. The terrain near the edges will probably be a bit ragged. If that's the look you're trying to achieve, that's fine, but more than likely you'll need to smooth it out with the aid of the **smoothing tool**.



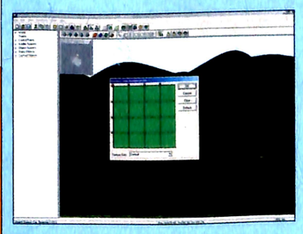
05 FIXING THE MATERIAL

Switch to the **Material Mapper** and you'll quickly see the new material that has been added. It seems the tool will always make the road wider than it should, so adjust it to closer fit the type of road that you are making. One grid in *BattleCraft* is approximately 4 meters in the game, which conveniently works out to be just about right for a one-lane road.



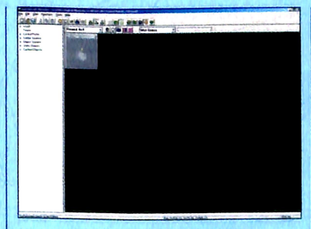
06 ADJUSTING THE MATERIAL

Think about how a real road looks; there's the actual road surface, then, usually, a strip of gravel or dirt, then dried grass or plants, and more healthy plants further away from the road. Of course, a road in the desert will be different from a road through a jungle. Adjust the material map to something similar. The material map you're adjusting here will be the basis of the terrain texture.



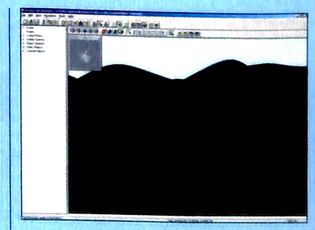
07 TEXTURING 101

After *BattleCraft* generates surface maps, they will look fairly generic. These are fine for preliminary development, but will need a bit of work before they look decent. Luckily, *BattleCraft* has a texture editor called the **Surface Map Painter**. Some people have reported problems with it, but if you select the texture set you used originally, any problems are minimised. Before using the **Surface Map Painter**, open **Surface Map Assignments** dialog and select the right texture set.



08 SELECTING BRUSHES

After you have the basic textures created, you'll need to tweak them. Click on the **Surface Map Painter** button. *BattleCraft* will prepare the textures for editing, and then you can go ahead and paint. There are several brushes that are supplied with *BattleCraft*, which are useful for carrying out standard texturing chores. Select the various brushes and paint them around the texture to see how they work. Use the intensity slider to increase the effect of the brush.



09 PAINTING MATERIAL

The **Paint Texture** on Material function is useful to tweak the textures that have been created from the material settings. Use it to give the textures a more life-like appearance. Give the edges of the road a ragged effect with the noise brushes. A cool feature of the painter is, if you paint over an area that you didn't want to, you can switch back to the original texture and paint over the mistake. The textures will always line up as long as they haven't been modified by you.

TESTING YOUR MAP

Get it right first time with test map function.

Although *BattleCraft* has a test map function, it's not essential that you use it, so consider whether you want to bother running it first; is it worth it just so you can show your map to some friends? If so, you can run your map on a local server that you create. Here's how: *Battlefield 1942* and *Battlefield Vietnam* are pretty much the same: run the game and select your profile, if needed. Select the Multiplay option and then Create Game. Find your map on the list and double-click it to add it to the server list. Hit the Start Local button and your map will load.

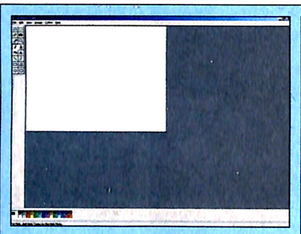
PAINTING BY NUMBERS

It's all about picture perfect pixels.

The bitmaps that *BattleCraft* uses to paint with are all 8-bit grayscale images. The level of black in each pixel determines how heavy that pixel paints. Black does not paint, and white paints solid, according to the intensity slider setting. Gray tones can be used to achieve various shades in between. You can achieve various effects by manipulating how dark and light the pixels in the brush are.

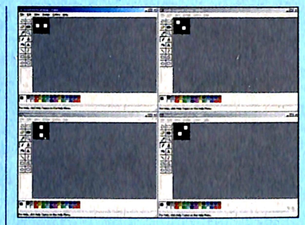
// CREATING NEW BRUSHES FOR BATTLECRAFT

Create a brush for practically any purpose.



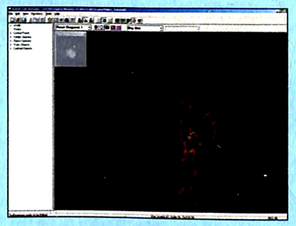
10 OPEN YOUR IMAGE EDITOR

Ever wanted to make a perfectly aligned two-track road? It's not easy using the single brush strokes supplied with *BattleCraft* to do this. What you need is a custom brush. You can create brushes in any image editing program like *Adobe Photoshop*, *Corel PhotoPaint* or *Jasc Paint Shop Pro*. But, you may not have these. A stripped down image editor included with Windows called *Windows Paint* can be used to create brushes. Find it in the accessories section of your start menu.



11 MAKING A NEW BRUSH

Open the **Line 8x8.bmp** which can be found in the **Brushes folder** where you installed *BattleCraft*. Use the tools in your image editor to change the small bitmap so you have one 4x4 pixel square offset from the other by one pixel. This will give you a brush that will allow you to create a road track that can turn corners in one direction. Save it with a unique name that signifies what it does. Make a set that can cover any angle road that you think you'll need.

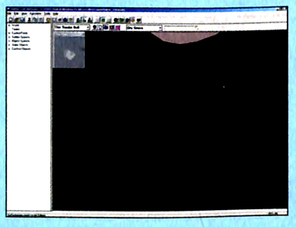


12 CREATING A TRACK WITH THE NEW BRUSH

Open your map in *BattleCraft* and select the new brush you created. If you draw a vertical line and then turn a corner, you will see how the brush can create a set of parallel tracks that look like they were created by vehicles driving through the grass. Make a set of brushes that have the white pixels arranged so that you can create any type of track you want. If you make a mistake, simply paint over it with the base texture.

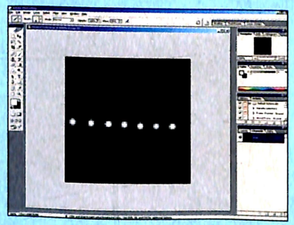
13 TIRE MARKS ON TRACKS

Use the same brush on a paved road to simulate the marks that vehicles would make with their tires. Use the brush in darken mode (the gray light bulb) with the intensity slider at about 35 per cent. This is a very subtle effect; you don't want to overdo it. Keep the effect spotty: don't fill in a solid line on the road surface. With the intensity slider at a lower setting, the faster you move the brush, the less effect it has.



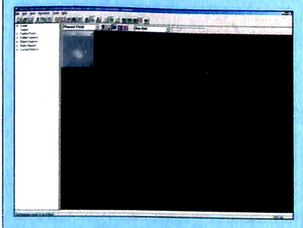
14 ONLY TIRE TRACKS

To create a texture that looks like it was never a road, but has been driven on by many vehicles, just use a small, one or two pixel brush and paint with a similar texture to the base texture. In the screenshot, you can see the base texture is wet grass. If enough vehicles travel the same route, the grass will die in the places it has been driven on the most. Use dry grass over wet grass, dry dirt over wet dirt, etc.



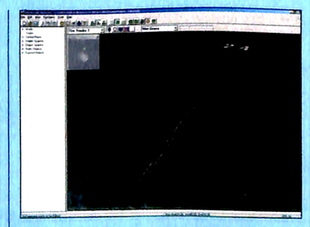
15 PLOWED FIELD BRUSH

Go back to your image editor and open the **Round 64x64 *BattleCraft* brush**. *Windows Paint* may not be the best tool for this job, but you can use it if that's all you have. A more capable image editor will give you better results with more ease. Create a series of small dots across the image, roughly in a row; don't spend too much time on this, they don't have to be perfect. Save it as **Plowed Field** and open *BattleCraft*.



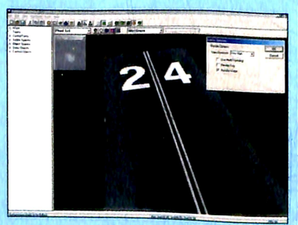
16 CREATE THE FIELD

Select your new plowed field brush and, using one of the dirt textures, paint in a straight line across a flat part of your map. A series of lines will appear, in perfect parallel formation. Mix two or three types of textures with different intensity settings to achieve a 3D effect. Experiment with darkening one side of the textures to simulate a shadow from the sun. Note which direction the sun would come from and darken the opposite side of the furrows to simulate a shadow.



17 CREATING AN AIRSTRIP

Creating a runway is easy to understand: just paint the ground with the colour or texture that best fits what type of runway it is. It can be high-tech and paved, or a dirt strip hacked from the jungle. Once you have created the runway itself, add decorations and markings. Use the Tire Tracks brush in darken mode to add markings left by the airplane tires during takeoff and landing. You can also create a custom brush that adds text or numbers.

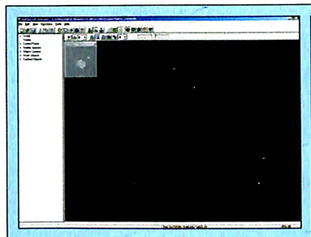


18 MULTI-TEXTURING MODE

BattleCraft normally displays the map in multi-textured mode, since this is how it will appear in the game. You can turn off multi-texturing in the editor options dialog by hitting **CTRL⇒E**. This turns off the detail texture that the map uses and instead simply displays the surface textures. You can use this mode to fine tune textures so that they align properly, and to see how much a dark or light brush is actually affecting the texture of your new map.

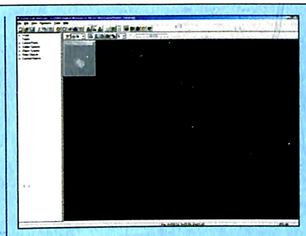
// INCOMING! CREATING BOMB CRATERS

In a combat zone, craters left by high explosives would be common.



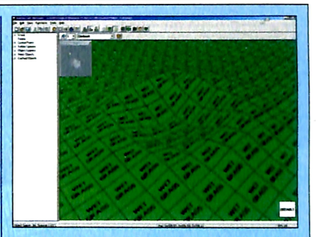
19 CREATING A CRATER

Switch *BattleCraft* to **Terrain Mapper** mode and select the **Raising Tool**. Change the settings for tool size and intensity to something that would match the size of the crater you're trying to make. Create a mound as tall as it is wide at the base. This will become the 'rim' of the crater. This step may be skipped if you're looking for a crater that is flatter or more depressed in shape.



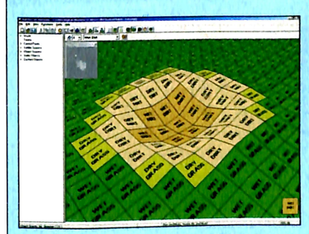
20 DROPPING THE BOMB

Switch to the **Carving tool** and decrease the tool size by one or two steps, depending on the crater shape that you're trying to achieve. Use it to sink the top of the mound you made in step one, until it's a decent depth. You will have to go by instinct here. Make it too deep and it will look out of proportion. Use the **smoothing tool** to finish off the rough edges.



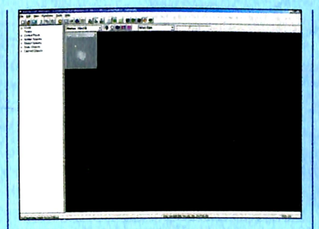
21 MORE MATERIALS

Once you have the crater made, it's time to assign a material set that would match what type of ground the crater is in. Switch to the **Material Mapper** and the default set of materials will display. Unless you know exactly where all your craters are going to be, the materials will need to be changed. Think about what type of ground the crater is in: will it be muddy at the bottom, or will it be dry sand?



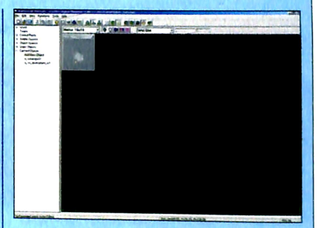
22 SETTING THE MATERIAL

A crater freshly dug into the ground is going to have more damp soil at its bottom, and disturbed earth near the rim, and perhaps some dead or burned vegetation around that, until normal vegetation takes over. Assign materials that will give the players accurate feedback as to what surface they are walking on. Once the players near the edge of the crater, the ground becomes soft and heavily soaked with the wet dirt that has been thrown up by the explosion.



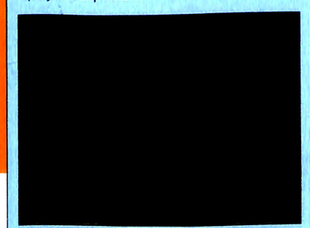
23 CUSTOM TEXTURING

You don't want to generate surface maps every time you create something new in the landscape. Either carry out all your terrain editing first, then create surface maps, or paint in custom textures. Open the **Surface Map Painter** and you will see the default textures, or whatever has been already painted there. Paint in the various textures that match the type of ground the crater is in. Blacken certain areas to suggest the heat of an explosion.



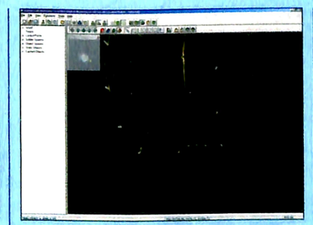
24 BOMBING STRUCTURES

To give the impression that a building has been bombed, add some craters around it. Place more than one to suggest a plane dropped several bombs in a row, some of which hit the target. Apply the same technique to roads, creating giant potholes for the players to avoid, or become stuck in. Unless you're making a map that represents a battle where huge amounts of bombing has taken place, a few bomb craters will go a long way: don't over do it.



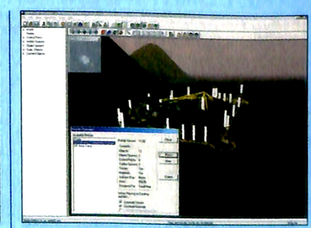
25 MUDDY CRATERS

In a jungle, any depression fills with water. A crater may also still be smoking. *Battlefield Vietnam* offers both options; *Battlefield 1942* only has the smoke. Go to **cached object list** and double-click on **add new object**. Search for **o_watermesh_m1** and **o_buildingsmoke** and add to your list. In *Battlefield 1942*, the object available is **buildingsmokeidledark**. Place these objects in the craters for added effect. The effects won't show up in *BattleCraft*, but will in game.



26 CREATING A PREFAB

Prefabs are great for placing repetitive sets of objects, and can be a good time-saver. When used incorrectly, though, they can give your map a 'cookie-cutter' appearance, so use them sparingly. Open the **Prefab manager** by choosing it from the **menu** or by hitting **F12**. To create a prefab, hit the **New button** and the editor will change to top-down view mode. Use the mouse to select the area that you have chosen to be in the new prefab.



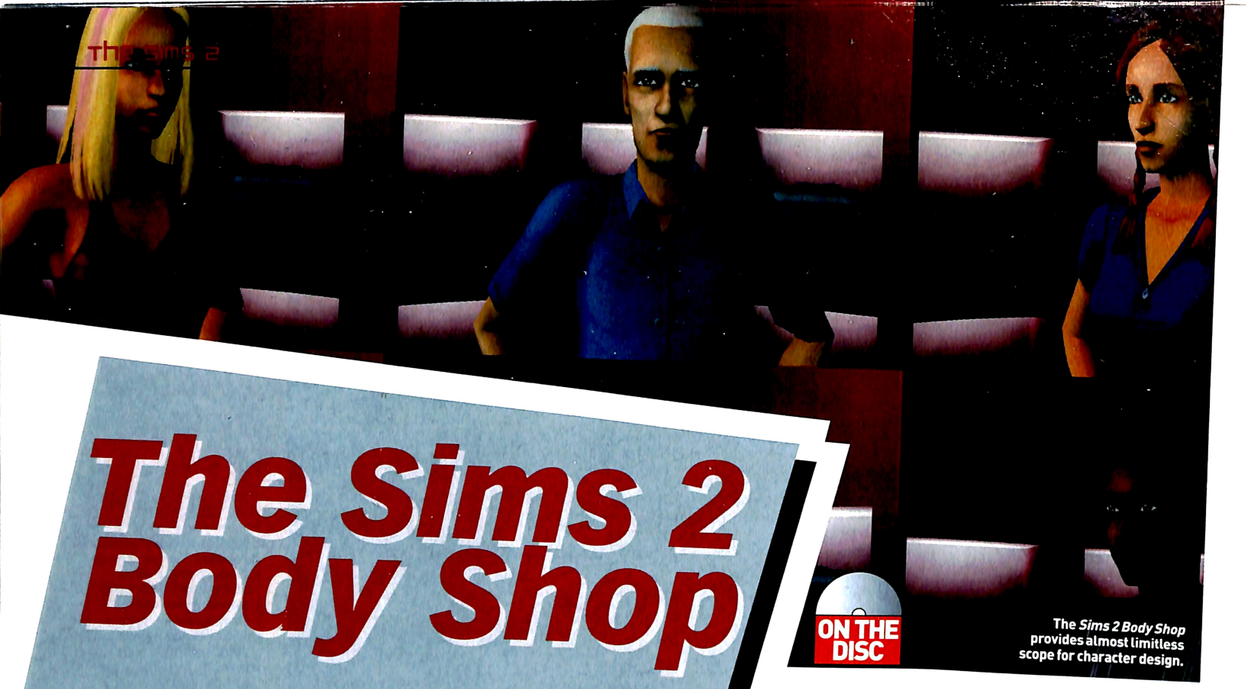
27 SAVING AND PLACING A PREFAB

When you are done selecting, the editor will ask you if you are satisfied, and then tell you the prefab has been set. Hit the **Create button** and give the prefab a name. You may also want to select options to save and place terrain and materials, depending on how you want the prefab saved. To place a **prefab**, select the name from the list, **hit Place** and point to the spot on the map where you want the prefab to be.

FILL UP WITH FOG

Matching fog settings with the skybox.

In both *Battlefield* games, you want your fog settings to match your skybox. *BattleCraft Vietnam* will ask you if you want to change your fog settings when you pick a new skybox. *BattleCraft 1942* currently does not. When you are creating a new map in *BattleCraft 1942*, open the map that contains the skybox you wish to use, and make a note of its fog settings, including colour. Then, open your new map and change its fog settings to match. Unless you are looking for a specific effect, matching fog to the skybox will ensure a smooth transition from in game atmosphere to the skybox graphic. If you are using a custom skybox, just open one of its graphic files in an image editor, and use the eyedropper tool to find out what colour the most common part of the sky is, using that for your fog setting.



The Sims 2 Body Shop

ON THE DISC

The *Sims 2 Body Shop* provides almost limitless scope for character design.

TRY YOUR HAND AT GENETICS AND START CREATING LIFE FOR *THE SIMS 2*, MONTHS *BEFORE THE GAME'S RELEASE*!

One of the most popular games of all time, *The Sims*, is currently being revised, ready to be released later this year as *The Sims 2*. Rather than being another expansion, the whole game has been completely overhauled and an eager fan base is currently chomping at the bit to get their hands on it. In a way to appease all you Sim-addicts, EA has recently released *Body Shop*. The stand-alone software, which is found on this month's cover-mounted DVD, is a version of the Create A Sim feature that will be part of the full game.

Modifying your Sims in the original was great fun and, as before, you are also able to kit out your Sim with an array of clothing and accessories: some of which you can customise. However, previously, it was tricky to simulate your Aunty Bella's hooked pigeon nose, or her wonky ears. Now *Body Shop* gives you the power to create a multitude of face types, ranging from the most beautiful to the downright ugly. What's more, *The Sims 2* game will recognise any genetic traits, so with a bit of tweaking, you can ween out that nose from future generations.

NICE PACKAGE

What to do when you have created your perfect Sim.

So, you've spent hours tweaking noses, shaving bits off jawlines, and generally perfecting your Sim. You've created a brand new wardrobe that will wow the pants off the greatest designers in Milan. With the release of the game still in the future, you now need a way to show the world your creations before you explode. Well, no need to worry: *Body Shop* is fully equipped to package up your artificial automatons so you can send them to friends or upload them to *The Sims 2* website.

In the Build Sims section of *Body Shop* you won't have failed to notice the section called Package Sims. Selecting that will take you to your Sim Bin where you can choose to either pack up your Sim to a file ready to load into *The Sims 2* or send it directly to TheSims2.com. Beforehand, give them a name and a little description and then pack away! One last thing to note is that, in order to send to the website, you must already be a registered user.

Neil Rennison is a Graphics Artist with experience in making games and has recently completed working on two racing titles for PS2, Xbox and PC, in which he provided the 3D artwork and track designs. His favourite games are historical based.

DIFFICULTY BASIC

TIME TAKEN 1 HOUR

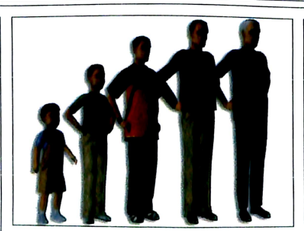
WHAT YOU NEED...
The *Sims 2 Body Shop* and the Starter Content Pack on your cover DVD.

// THE FUNKY INTERFACE

Finding your way around *Body Shop* is easy, and the interface is exciting too!

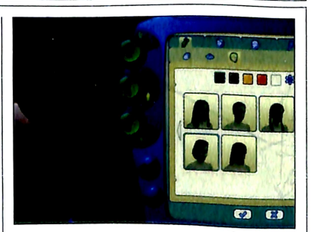
01 CLONING AROUND

Using the *Body Shop* is very straightforward. It is essentially split into two parts: **Build Sims** and **Create Parts**, which speak for themselves. For now, select **Build Sims**, and then select **Build** or **Clone Sims** in the proceeding menu. This will then take you into the **Sim Bin**: a storage area for future Sims. Click on the cross to create a new Sim!



02 AGES OF MAN

Firstly, a randomly generated Sim will appear on the left. Use the arrows at the bottom to spin them around. Choose the sex of your Sim: by selecting the **gender icons** and then select the age. The age shows you exactly how your Sim will look at various stages in their life. You can even decide if your Sim is slimline or of larger proportions.



03 FRENETIC GENETICS

Now you have decided on the basics, it's time to play geneticist! Click on the **Genetics icon** and you have three options to change: **skin tone**, **eye colour**, and **hair colour**. The hair section also allows you to change the hair from a swatch of styles. Later on, this tutorial will show you how to make your own custom colour swatches.

// EYE OF THE BEHOLDER

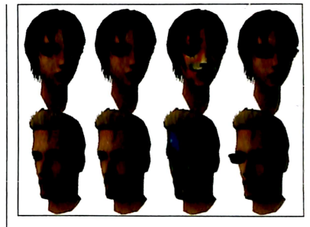
Body Shop enables the Michelangelo in you to create veritable beauties or unfortunate freaks.

04 FACE SCULPTING

Body Shop's most powerful tools are those that enable you to create an almost infinite selection of face types. Click the **Faces icon** and you will be able to change certain areas of the head such as the nose, jawline, or mouth, by using sliders to add maximum or minimum effects. The wealth of options is enormous and you should be able to create the Sim of your dreams.

05 PLASTIC SURGERY

The **Modifiers** are great fun to play with and can be used to provide small tweaks to perfect your Sim or experiment with the extremes of facial types: your very own big-nosed Sim-esque Cyrano de Bergerac! (if that's your thing). As before, there are different sections and each tweakable part has its own informative diagram to help you along the way.

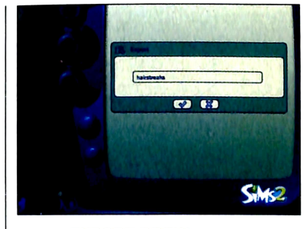


06 FASHION ACCESSORIES

The **Facial Hair**, **Makeup**, and **Glasses icon** does exactly what it says on the tin and allows you to give your give your Sim a variety of mods that will improve their street cred. Clicking the **Clothing icon** will further give your Sim a makeover and allow you to dress them in a variety of clothing types. It even allows you to mix and match top and bottom halves, to a certain degree.

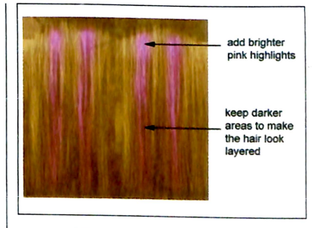
07 CREATING CUSTOM PARTS

Once you have exhausted the standard Sim parts, you will want to create your own. Click on the **Create Parts icon** and for this part of the tutorial, select **Start New Project** and then select **Create Genetics**. Also make sure the mannequin is a female adult. We are going to give her some funky hairstreaks, so select the **Hair icon**. Now select **Blonde** and the first **hair type**.



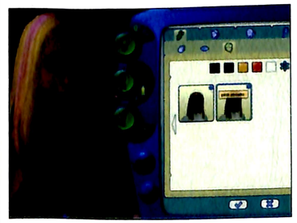
08 EXPORT BREW

Click on the **Export Selected Textures icon** at the bottom (the folder with the arrow). Type the project name as **Hairstreaks**. A project folder will now be created in **My Documents/EA Games/The Sims 2 Body Shop/Projects/** and all textures relating to that hairstyle will be available to you, including all the alpha maps. It will also save all five-age versions of that style.



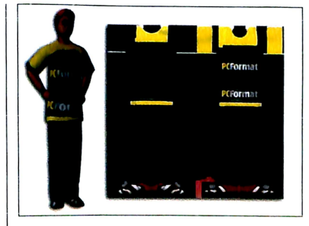
09 STREAKY BACON

Keep *Body Shop* open and, using an art package such as *Paint Shop Pro*, open up the texture named **afhair_Casual~hair alpha7~stdMatBaseTextureName.bmp**. Now using your amazing artistic ability, give the hair four vertical pinkish streaks. Try and make it look as natural as possible with dark areas and highlighted strands. Save the texture over the old one.



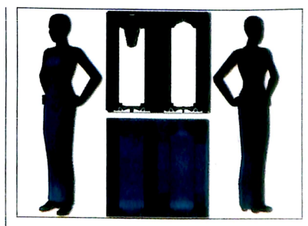
10 UPDATE THE FILE

Back in *Body Shop*, click on the **Refresh Preview Sim** (circular arrow icon) and the texture will appear magically on the mannequin. If you are happy with it, type **Pink Streaks** in the **tooltip box** and select the **Import to Game icon** at the bottom. Now build a new Sim and when in the **Hair section of Genetics**, select the **Custom icon** to choose the pink streaks style!



11 FASHION DESIGN

Now you have customised one aspect of *Body Shop*, you will be able to have a go at customising other Sim-parts such as beards, eye and skin colour, or everyone's favourite: clothing. It may take some testing to make sure the textures work, but you should be able to create stunning new fashions for your Sims (like this trendy *PCFormat* shirt: next season's essential.)



12 ALPHA FUN!

Once you gain confidence in designing new clothes, it may be worth looking at modifying the alpha textures to generate new effects. Any part of the texture that is black will be transparent and any part that is white will show the main base texture. You can easily customise a dress within the alpha texture, as shown above. Notice how the straps only appear in the alpha.

SIM NEWS

Gagging for more info on *The Sims 2*?

» In the game, each Sim will not necessarily need to wear the same clothes all the time. In the game there will be a mode called *Create A Family* which will enable you to have more clothes to choose from.

» Each generation will last a different amount of time. The toddler stage is the shortest and the adult stage the longest.

» Each generation will also have its own requirements in order to keep the Sim living healthily and achieve a decent life score.

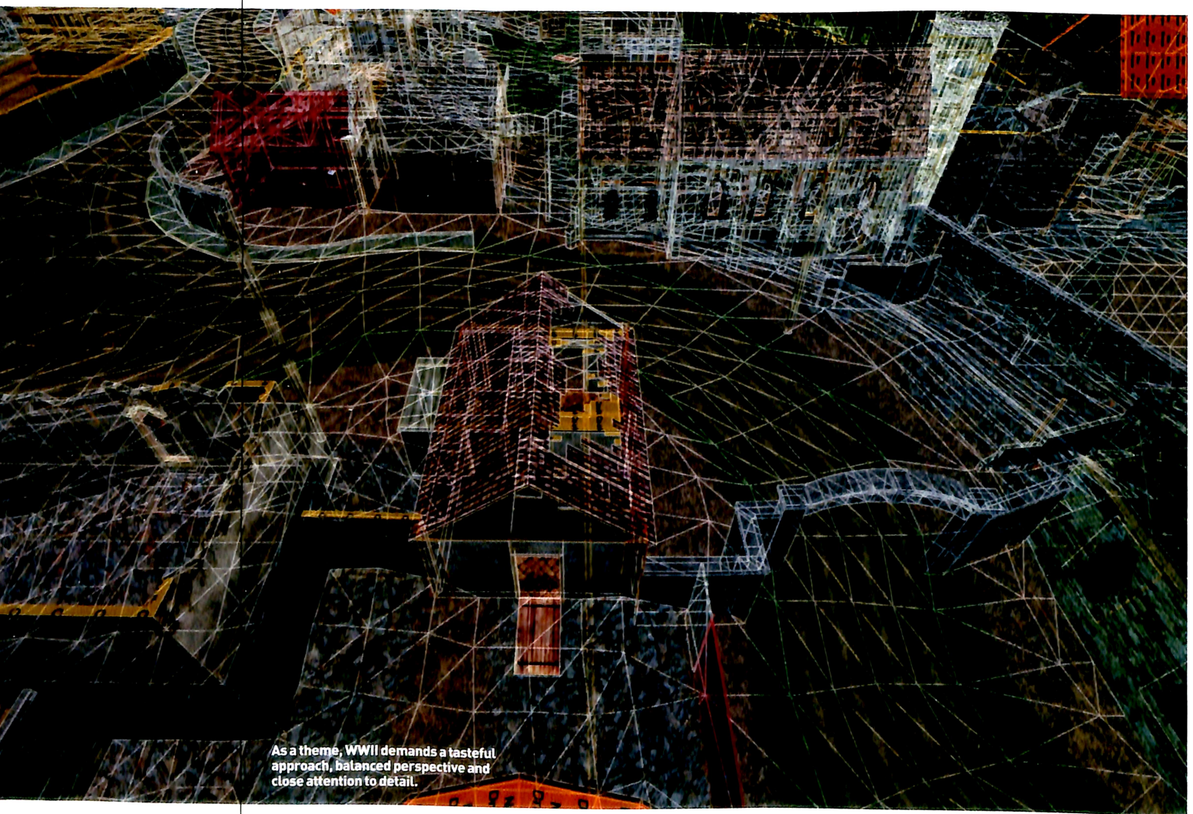
» Life scores will be used to assess the success of the Sim's life and will determine what is available to the Sim throughout their life. A child with a bad upbringing will have a low life score, which may harm their career and love prospects!

» It is possible to turn off the ageing process within the game – but surely that's no fun!

PART ONE

Call of Duty MAPPING

THE TIME HAS FINALLY COME TO WAGE YOUR OWN WAR IN *CALL OF DUTY*. READ ON AND DIG IN FOR A LONG CAMPAIGN.



As a theme, WWII demands a tasteful approach, balanced perspective and close attention to detail.

ON THE DISC

Call of Duty is based upon the versatile *Quake III* engine and consequently uses *Radiant* as its map editor of choice. So if you are familiar with mapping in *Jedi Knight 2*, *Medal of Honour* or indeed *Quake III*, these tutorials won't reveal a great deal to you. However, reading this tutorial will teach you how to map for all four games, although naturally each version does have their very own idiosyncrasies.

CoDRadiant is your typical low polygon world builder, featuring multiple view points and a 3D view for easy navigation, a texture editor to paint your world with and various other gizmos to make map creation a relatively painless experience.

We will assume you have no past experience with any 3D packages, so will begin with the very basics. If you get stuck on any step of the tutorial, you can refer to the examples found on the cover disc. Included with these are the batch files needed to build each map. You will need to copy the .map files over to your **Call of Duty\Main\maps** folder before attempting to compile them. Read the section on batch files before attempting to run them.

Nick Mohr has been actively involved in the maps and modding community for the last 6 years, creating numerous maps, models and textures for *Quake*, *Team Fortress Classic* and *Day of Defeat*. He's currently working on an exclusive top secret mod for *Half-Life 2*.

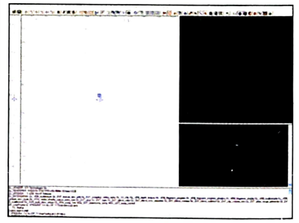
DIFFICULTY INTERMEDIATE

TIME TAKEN 2 HRS 30 MINS

WHAT YOU NEED...
CoDRadiant – available on the cover DVD (codtools.exe).

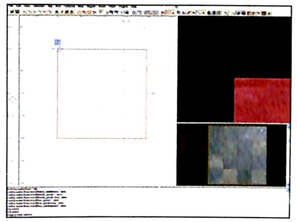
// BRUSHES AND TEXTURES IN CODRADIANT

The fundamentals of any level begin with brushes and textures. Here's how to build them.



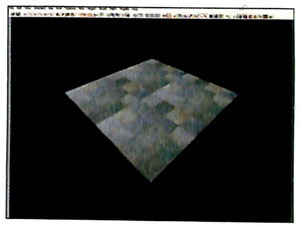
01 INSTALLATION AND SETUP

CoDRadiant installs by default to your *Call of Duty* folder which enables it to find all textures and assets it needs to build a map. The required **cod.qe4** file can be found within the tools/bin folder. This is needed to correctly set up paths crucial to the program; do not edit this file unless you know what you are doing. If *CodRadiant* ever fails, or behaves strangely, you can always use the registry editor to delete the **HKEY_CURRENT_USER\ Software\ CoDRadiant** folder.



02 BRUSH CREATION

Brushes are the building blocks of all maps; they are the name given to any geometry that is created in the level. Before creating any brush, an appropriate texture should be chosen. In the editor, click **Textures ➠Austria ➠Floors,** and find **rock@floor_granite1** in the texture preview window. In the **XY plane**, left click mouse and drag out a square 256x256 units in size. You can read the dimension of the selected brush from the left and bottom sides of the brush.



03 NAVIGATING CODRADIANT

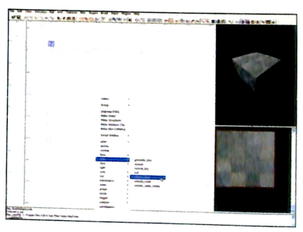
Right clicking and dragging the mouse in the **XY plane** allows you to pan, while the mouse wheel lets you zoom. In the 3D view, navigation is controlled by cursor keys: the **[D]** and **[C]** keys allow you to move up and down whereas **[A]** and **[Z]** control the vertical angle you are looking at.
Selecting and deselecting brushes is controlled by **[Shift] ➠Left mouse button**; to deselect multiple objects, hit **[Escape]**. **[Backspace]** will delete any selected object.

04 THE FIRST ROOM

Now we need to create the walls and a ceiling. Deselect the first brush and make a new one, 8x256 units. If needed, the grid size can be changed using keyboard number **[1]** to **[9]**: the grid size is always shown in the bottom right of *CoDRadiant*. The height of the brush can be adjusted in the **Z plane** or by changing the default **XY view** to **XZ/ YZ axis** using the change view button **[CTRL] ➠[TAB]**. Make the new brush 192 units tall.

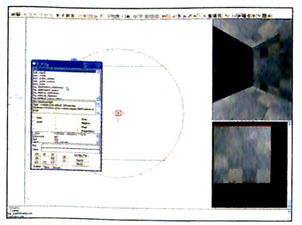
05 ROTATING BRUSHES

Clone this brush by selecting it and pressing **[Space]**, and drag it over to the other side of the floor. Check in the **XY** and **XZ** views that there is no space separating the walls and floor. Clone one of the walls again and rotate; **click Selection ➠Rotate ➠Rotate Z**, align it against the floor and repeat for the final wall. Now build a ceiling using the same method, cloning the floor and realigning it at the top. Make sure you leave no gaps.



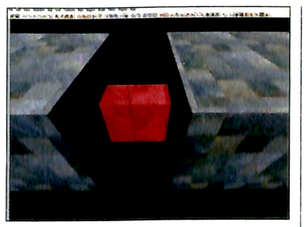
06 BASIC ENTITIES

Entities are objects added in the map to give your world interactivity and a sense of realism. They are used for lights, sounds, doors and just about anything that is not a part of the solid geometry. Before we can compile this map we need to add at least two entities, a light, and the player's start position. Right click in the main plane and select **info\player_ start**, and position the entity in the room making sure it's not intercepting any geometry.



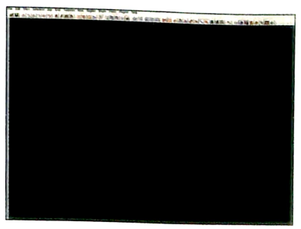
07 ENTITY PROPERTIES

A light can be added in the same way as before. Right click and select light then reposition the entity in the room. With the light highlighted, press **[K]**, and select any colour you wish (light yellow is normally the most suitable). Press **[N]** to bring up the entity properties dialog and you will see the light colour has already been added. To change the brightness, enter **light** as the **Key** and any number as a Value: 300 is default. You can now compile the map.



08 CSG SUBTRACTION

Subtraction is a convenient way to carve out holes in geometry to make doors, windows and other features. Create a whole new room by selecting all brushes, then clone it and drag it to one side and deselect it. Now you can create a new brush that will represent a virtual corridor connecting the two rooms; make sure the brush intersects the facing walls of each room so doorways can be created. Allow the doorways you have created to be 64x112 units in dimension.



09 JOINING ROOMS

With this virtual corridor now created and highlighted, click **Selection ➠CSG ➠ Subtract** which will carve out any brushes that the selected item was intersecting. Now hit **[Backspace]** to delete the bush. This is the easiest way to create doorways or windows, unless you want to build the individual brushes yourself. You can now build new brushes to create a corridor and connect the rooms. Again remember to make sure there are no gaps between the rooms to avoid leaks.

BATCH FILES AND COMPILING

Compiling is not complicated but you must check your paths.

On our cover disc are the batch files needed to compile the levels: called step1.bat etc. Double click the batch file to compile its associated map.

Compiling takes place in three stages: BSP creation, visibility creation and lighting. The first two are performed by q3map.exe and the latter by flare.exe. Open a batch file with notepad and examine the directories, by default both q3map.exe and flare.exe can be found in C:\Program Files\Call of Duty\Tools\bin.
If you installed *Call of Duty* and its tools to another location, you will have to change these paths. The second path on each line points to the map you are compiling; you should always save your map in *CoDRadiant* to the maps folders, by default C:\Program Files\Call of Duty\Main\maps, so that when they compile they can be found by *Call of Duty*. Change these paths if you did not initially install the game to the default location.
To run your map, load up *Call of Duty* in single player mode, pull down the console using the [`] tilde key and, in the console, type map step1 etc.

LEVEL DESIGN PHILOSOPHY

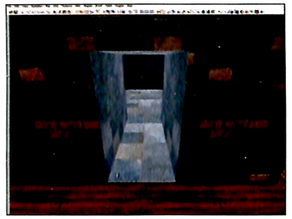
Don't get too bogged down with detail.

Everyone likes an attractive map with rolling hills, pretty fountains and exquisite architecture, detailed down to the last rock in a crumbling wall. It immediately draws the player's attention, but the question is, will it hold their attention? Gamers are an extremely fickle lot and eye candy is not the be all and end all of a game. Your focus as a map maker should begin and end with gameplay; there are a few simple rules to remember when you are designing and creating a map.

All maps have objectives. A player should immediately know what they are doing and what role they have to play in the game: for example a castle should be a base of defence and not the platform of an attack and a bomb site should have something in it that deserves to be blown up. In a hectic multiplayer game, the player will not be looking at the lamppost you painstakingly made, but instead will be looking for places to pin down the enemy or get them caught in crossfire. Think about how a player will move, where they will hide and how they will use their surroundings. Most importantly, make your map fun.

// DETAILING YOUR WORLD

You can make an interesting level with more complex geometry and entities.

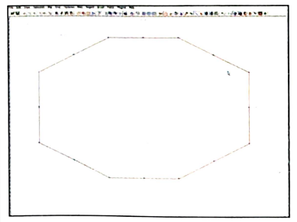


10 TEXTURE CONTROL

When texturing brushes, you may find they are not aligned right or you want to apply a texture to just one face of a brush. To select an individual face, press **[Ctrl]➡[Shift]➡left mouse button in the 3D view**. Hit **[S]** to bring up the surface inspector. From here you can shift, stretch, rotate and auto fit the selected texture: it is all quite self explanatory. To change the texture, left click on your desired choice in the texture preview plane.

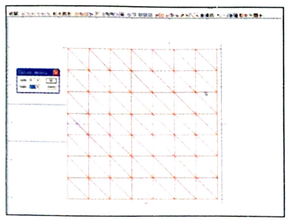
11 CLIPPING TOOL

There are a number of ways to modify brushes; one of the most powerful is the clipper tool that allows you to slice parts off a brush. It lets you round edges or creates a wedge out of a cube. Highlight the brush you want to clip and hit **[X]**. Clicking on two points in the **XY view** will create a virtual knife; to swap the selection, press **[Ctrl]➡[Return]** or press **[Return]** to perform the operation. Brushes can be clipped along any axis.



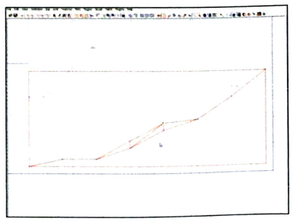
12 EDGE TOOL

If you created a new shape using the clipping tool, you'll find that dragging the edges only scales the brush. You can further manipulate the brush using the **Edge tool**. Hit **[E]** to start it while a brush is selected, and see points highlighted along the edges and vertices. Stretch the edges by selecting the points on the corners. Take care when manipulating brushes; complex shapes can cause compile errors.



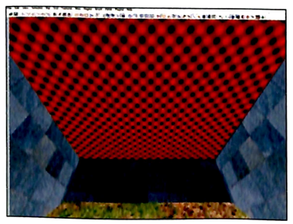
13 TERRAIN MESHES

Terrain can be created out of any flat brush. Build a new brush and click **Patch➡Simple Terrain Mesh**. The terrain density dialog is used to describe how detailed you wish the terrain to be: change both width and height to 8. Hit **[V]** to start the **Vertex Tool**. As Now, you can see that individual points can be dragged in any direction: it is easiest to do this in the **2D views**, selecting a point in the **XY plane** and then altering its height in the **XZ plane**.



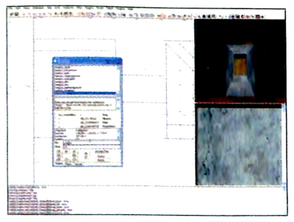
14 ROUGHING IT

Selecting a point whilst pressing **[Shift]** will highlight a whole axis, making it easy to uniformly change the height; clicking the point again will change it to the perpendicular axis. You can add small undulations by selecting individual points or using **[Ctrl]** to group multiple points, and then **[Escape]** to deselect all points. It is sometimes better to create terrain in a separate map than import it: this lets you quickly and easily identify any problems.



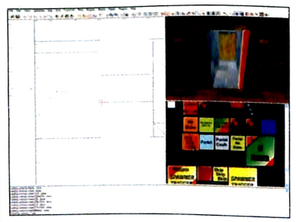
15 TO THE SKIES

Sky is applied like any other texture in the map. The difference is that you will not see the texture appear in *CoDRadiant*; instead just as a strange red and black pattern. Select the brush you wish to use as sky then bring up the Surface Inspector by pressing **[S]**. Type **sky/stormy** as the texture value and click apply. There are many sky textures available, but some good examples are dam, chateau and airfield. You can find more skies in the Pak files discussed later.



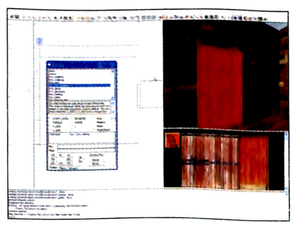
16 WORLDSPAWN ENTITY

By default, any sky will illuminate the world; you can change the colour, brightness and angle using the **worldspawn entity**. This entity is automatically included in all maps and can be accessed by pressing **[N]** , and selecting **worldspawn**. From here you can add values for suncolor, sunlight and sundirection to change the colour, brightness and direction. You will notice the **Entity Properties editor** has descriptions included for all the different values you can change.



17 MAKING DOORS

Normal sliding doors that you would find in games like *Quake III* cannot be practically implemented in *Call of Duty* without complex scripting, so we have to stick a more traditional rotating door of the World War Two era. To create a rotating door, two brushes must be used. The first is obviously a brush that is textured to represent the door and the second is used as the point of rotation; this second brush must use the origin texture from **Textures\Common**.

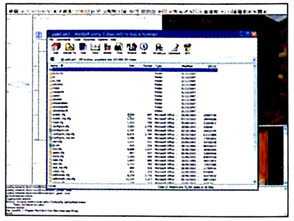


18 MORE DOORS

Build these two brushes and place the origin brush at the point of rotation: think of it as the hinge. You now need to group the two brushes using **[Shift]➡Left mouse**, now press **[N]** to bring up the **Entity dialog** and double click on **func_door_rotating**. From here you can add new keys and values to change the speed and angle it opens at. Notice that using the **teamname key** with the same values allows you to be able to make double doors.

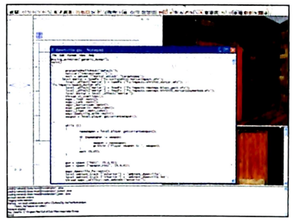
// CREATING INTERACTIVITY

Making more complex entities involves scripting the GSC files and further use of entities.



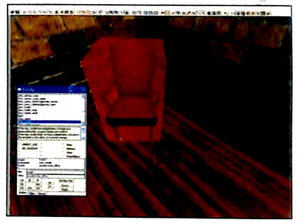
19 PAK FILES

Pak files are used to store the assets that *Call of Duty* uses. They are found within the main folder with the pk3 extensions and can be opened with programs such as *Winzip* and *Winrar*. When adding sounds or models, you will have to refer to these files to find the correct paths. *Winzip* by default does not show a directory structure but will display the paths in detailed view. We will explain how to point to individual files in the next few steps.



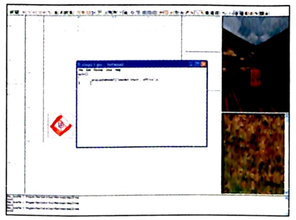
20 GSC FILES

These files are used in conjunction with the BSP files to tell the game how, when and where events should happen; they must be placed in the maps folder alongside the BSP of the same name. You can edit GSC files with *notepad*. Open the **dawnville.gsc file** and examine it; if you are familiar with **C** or **C++** you will see familiarities between the styles. All scripts must have a main() function with { and } surrounding all of the code.



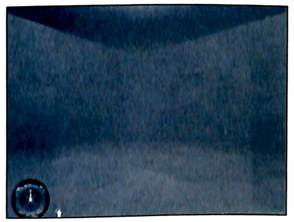
21 ADDING MODELS

Create a new entity in a level using **misc/model**, then click **Cancel** on the window that opens. Hit **[N]** to bring up its entity properties and add model as the key and **xmodel/chair_office** as the value. This model can be found in the **pak0.pk3 file** as discussed previously in step 20. In the 3D view you should now see a chair. The chair can be rotated from the **Entity Properties editor** and moved just like any brush. Save as step22.



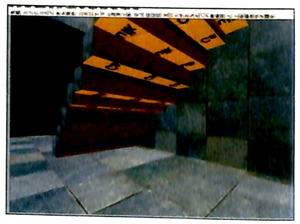
22 EDITING GSC FILES

Before you can run this map properly, the model has to be loaded into the game from the Pak file: this is where the GSC files come into play. A line must be added to the **step23.gsc file** to tell the game this fact. If you open the **step22.gsc**, you will already see the line **precacheModel ("xmodel/chair_office")**. For each model that is used, you must remember to add this command to the GSC file. The GSC files must reside in the map's folder.



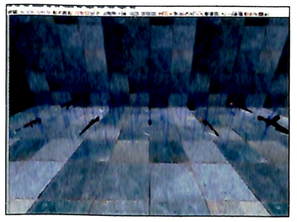
23 ADDING FOG

Fog can be added to the map by attaching the following line to your GSC file **setCullFog** (start distance, end distance, R, G, B, transparency). Start and end distance are standard integers, whilst the RGB values are between 0 and 1 to two decimals places. Simply take your normal RGB values and divide them by 255 to work it out. The transparency value should normally be set to 0 for best results. Check out **step23.map** and **step23.gsc** files to see how it works.



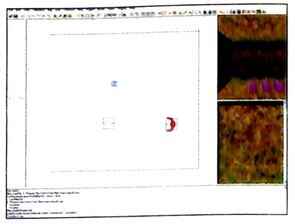
24 CAULK TEXTURE

Up until now you have been building brushes with preselected textures and whilst this is perfectly acceptable and has many advantages, using the caulk texture is preferable for any face of a brush that cannot be seen by the player. Caulking brushes lets the game engine know it does not need to render them, thus speeding up in game performance. An example of a suitable caulking location is underneath a staircase: inaccessible from the rest of the world.



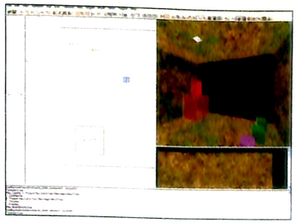
25 FIRE POWER

Supplying weapons in *Call of Duty* is incredibly easy: they are nothing more than predefined entities and do not even need scripting in a GSC file. Right click in the 2D plane and select **Weapon➡Any weapon** you want. From within the entity's properties, you can change the angle they point, suspend them in the air and let them respawn. If you want the player to have a certain weapon at the beginning of the level, simply put them under the **info_player_start point**.



26 ALLY AI

You have probably already worked out that you can add allies and axis players to the map but they do not seem to do much at all. To get them to interact with the environment there are a number of entities that need to be added. Firstly you must link the ally players to the **info_player_start** by highlighting them and pressing **[Ctrl]➡[K]**. Then create a number of **node\pathnode** entities which help the AI navigate the map, and place them in sensible positions.



27 TRIGGERING AI

Now we can create a trigger and an action for the AI. Create a brush over the **info_player_start** position using the **Common\AI Trigger** texture, then turn this into a **Trigger\Friendlychain** entity. Create two **node\cover_crouch** entities, or any action you want, and link them to the AI Trigger using **[CTRL]➡[K]**. Before you test this map, the game must connect the node paths. To do this, type **g_connectpaths 1** into the console before running the map.

THE PROBLEM WITH LEAKS

The most annoying and frustrating problems for modern mappers.

Leaks are the bane of every mappers life. A leak is basically any part of your map that has a hole into the void outside your world. The normal analogy is to compare your map with a bucket: wherever the water can escape, is a leak. Leaks happen for a variety of reason, normally because the map has walls that have not been properly aligned, but other reasons include an entity outside the world or overly complex brushes. When a map has a leak, there are a number of problems: it will not compile properly, a visibility matrix will not be formed and no lighting will be rendered. You'll know you have a leak when you see no lighting and shadows in your level. But luckily *CoDRadiant* has a very handy feature that allows for easy tracking and correction of leaks. When the compiler detects a leak, it creates a new file which points directly to the location of the suspected problem. This file can be loaded within *CoDRadiant* by clicking File➡Pointfile. You can track the leak in the editor by following the thick red line that appears.

Look forward to the past. ***Sherlock Holmes*** **is heading back to the PC, courtesy of The Adventure Company.**

WHAT THE BLENDER SAW

STEP INTO A WHOLE NEW WORLD OF DANGER, INTRIGUE AND KILLER KITCHEN APPLIANCES. WHEN WAS THE LAST TIME YOU GOT TO DO THAT IN FRONT OF YOUR COMPUTER? SAYS RICHARD COBBETT.

PART TWO

The Pygmalion Institute sits atop a high cliff, on a remote island somewhere in the midst of the Mediterranean. Inside, its scientists are hard at work on every type of project imaginable, from smart warheads to artificially intelligent robots capable of doing the Macarena. Sometimes the two combine. Sure, that means bad news for the area, although definitely a bonus for contemporary dance.

Needless to say, the Institute is no stranger to would-be cat-burglars. Like the one the dogs had for breakfast yesterday. Or the one in the main wind chamber, currently redefining Crash Test Dummy. Or of course, the one in the black suit, currently clambering up the side of the sheer rock wall, heading straight for a seemingly innocuous back-garden. You see, there's this one professor – renowned for being a bit of a recluse. No cameras. No guards. No problem.

That's how it always begins, isn't it? Still. What's the worst that could happen? Someone could say "What's the worst that could happen", for starters. But that'd be stupid. Nobody would ever invite Fate to boot them up the backside, just before the score of their career. No, this is going to be easy-in, easy-out. And nothing – man or appliance – is going to get in your way.

INFORMAL INFORM

Last month, we looked at some of the basics of Inform programming. We created a couple of simple rooms, played with some objects, and set up a quick puzzle. This month, we're going to get started on a fully-fledged game. This is going to be fairly simple, involving only a few small rooms, a couple of player characters, some basic intelligence routines and a couple of nips and tucks to tidy everything up.

What we're not going to do is simply start at the beginning and type out reams and reams of code. Instead, we'll be focusing on individual elements of the design process, using a few examples from the actual game. Once we've done this, you'll find it much easier to work through the complete code on next month's coverdisc.

Richard has been playing text adventures since the *Infocom* days. He knows his plugh from his frotz and 'aint afraid of no grue. His favourite IF game is *A Mind Forever Voyaging*, while the last he wrote himself was called *Shapes In The Sand*.

DIFFICULTY INTERMEDIATE

TIME TAKEN 2 HOURS

WHAT YOU NEED... Nothing. Not a thing.

Before we begin coding, we need to carry out a bit of design work. Our sample game is set in the Professor's house, on the edge of a cliff. The player begins in the back garden, and their first challenge is to find a way inside. They'll do this by clambering up onto the roof, courtesy of an inventory item. What they don't know is that all of this is actually a 'false start'. Having broken into the house by accidentally falling through a skylight, they find themselves in a completely different predicament – the Professor who owns the place is dead, the house is sealed by a paranoid, McCarthyist security system, and the only hope you've got of getting out is to solve the murder. Easy, yes?

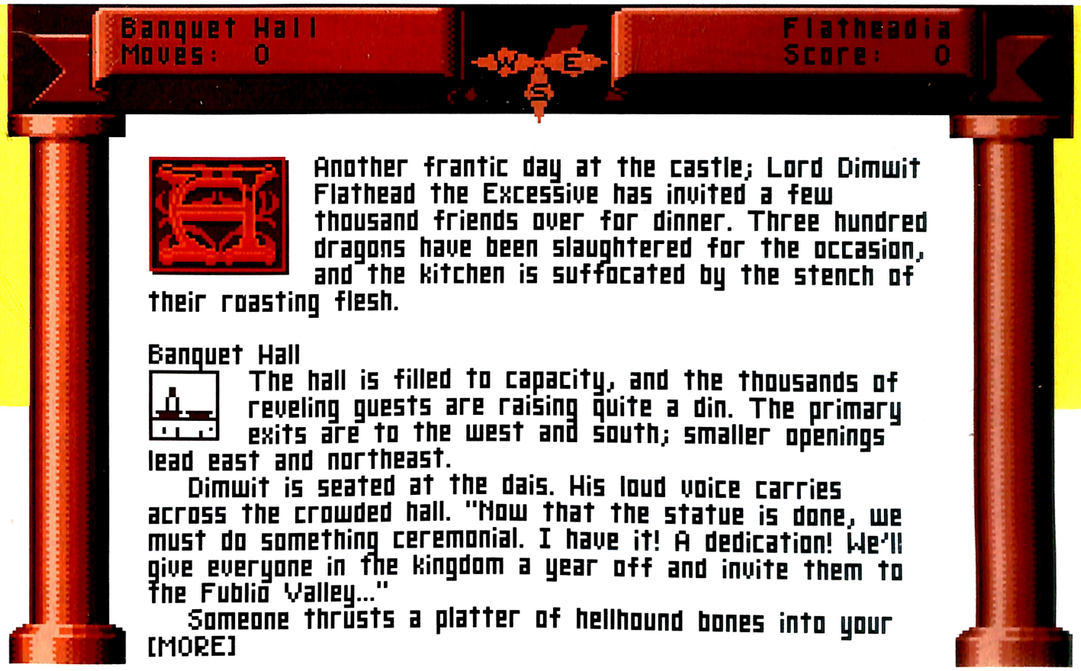
Oh, didn't we mention? The prime suspects are a food blender blessed with artificial intelligence, a garbage disposal unit, a toaster, a radio and the Professor's talking cat Cordelia.

IN THE MIX

Our first job is to work out exactly what the player's objectives are going to be. Clearly, in this case, that's 'get out of the house'. Let's start by breaking that up a bit. There are three rooms outside the house: the helipad, the back of the house and up on the roof. Inside, we have five more: the professor's kitchen, the living room (which you can't travel through until you've befriended the security system, Barrett Holmes), and a hallway – leading off into the professor's bedroom (where you find his corpse) – and an underground lab containing a partly built robot that you have to finish building, which finally lets you escape.

Our overall game plan is very straightforward. After waking up in the kitchen, the player has to find a way of impressing Holmes in order to access the hallway. Once they find the corpse, they discover a message in Cordelia's collar, instructing them to track down an informant called Deep Throat. This turns out to be the garbage disposal, which gives them the key for the underground lab. Once unlocked, they rebuild the robot, only to find that it has no brain. They fit the blender in its place, and it promptly turns on them. Once defeated, the endgame plays.

Let's break things down further into the pieces that we'll need. We now have a list of the key rooms that we'll need, each filled with the usual array of objects to poke and prod at. Amongst these, the most important are the Appliances – Citizen Blender, Deep Throat, Barrett Holmes and so on, each of which have to be converse with the player on a range of subjects. Citizen Blender also needs to be an item in his own right, so that we can move him into the right


Banquet Hall
Moves: 0
W E S
Flatheadia
Score: 0
Another frantic day at the castle; Lord Dimwit Flathead the Excessive has invited a few thousand friends over for dinner. Three hundred dragons have been slaughtered for the occasion, and the kitchen is suffocated by the stench of their roasting flesh.
Banquet Hall
The hall is filled to capacity, and the thousands of reveling guests are raising quite a din. The primary exits are to the west and south; smaller openings lead east and northeast.
Dimwit is seated at the dais. His loud voice carries across the crowded hall. "Now that the statue is done, we must do something ceremonial. I have it! A dedication! We'll give everyone in the kingdom a year off and invite them to the Fublio Valley..."
Someone thrusts a platter of hellhound bones into your
[MORE]

position for the endgame. Finally, for the characters at least, Cordelia is able to move around the house at will. To add a little variety, her presence in a room will affect many of our descriptions: sarcastic comments attaching to the player's actions and observations.

A ROOM WITH A VIEW

Last month's rooms were all uniformly simple. No description, no random objects, no unique features. Sadly, we can't get away with that in a full game. Let's begin by simply dropping the code for our starting room.

```
Object Helipad_Landing "Helipad Landing"
with name ,
description "You'd expected more, somehow. The top of
the cliff looks for all the world like your everyday back-
garden – albeit one with a God humbling view of the
Caribbean ocean, a helipad slap-bang in the middle of the
lawn and a little ornament in the shape of a cat, whose
eyes seem to be watching your every movement.
Checking your compass, you see that the front porch lies
to the north – although that's more professional interest
than dedicated orienteering. It's only a few meters away,
after all.",

n_to patio,
s_to "There's just the cliff that way, and it's a long way
down...",
d_to "It's a long, difficult climb back down to the boat, and
you're not doing that until you've retrieved the briefcase",
w_to "Sure, you could try to get over the wall, sneak your
way into the Institute via the standard thieves' entrances,
engage in all manner of tense stand-offs with the guards
and emerge victorious as the very symbol of your craft.
However, right now, staying alive is more of a priority.
There are guards and sensors and cameras everywhere
out there, while the researcher who owns this place
values his privacy. Much easier for all concerned...",
e_to "A giant, barbed wire covered wall prevents your
movement.",
u_to "You're already at the top of the cliff.",

has light;
```

Familiar? It should be. This is much more text heavy than our previous examples, but the actual code is precisely the same as before. Going North takes us to the next area, while the other directions print out a reason why the other directions are blocked off. In practice, this entire area does nothing but railroad the player into the house – but as you can see, we've gone to some lengths to disguise the fact. Every object that we mention has a description:

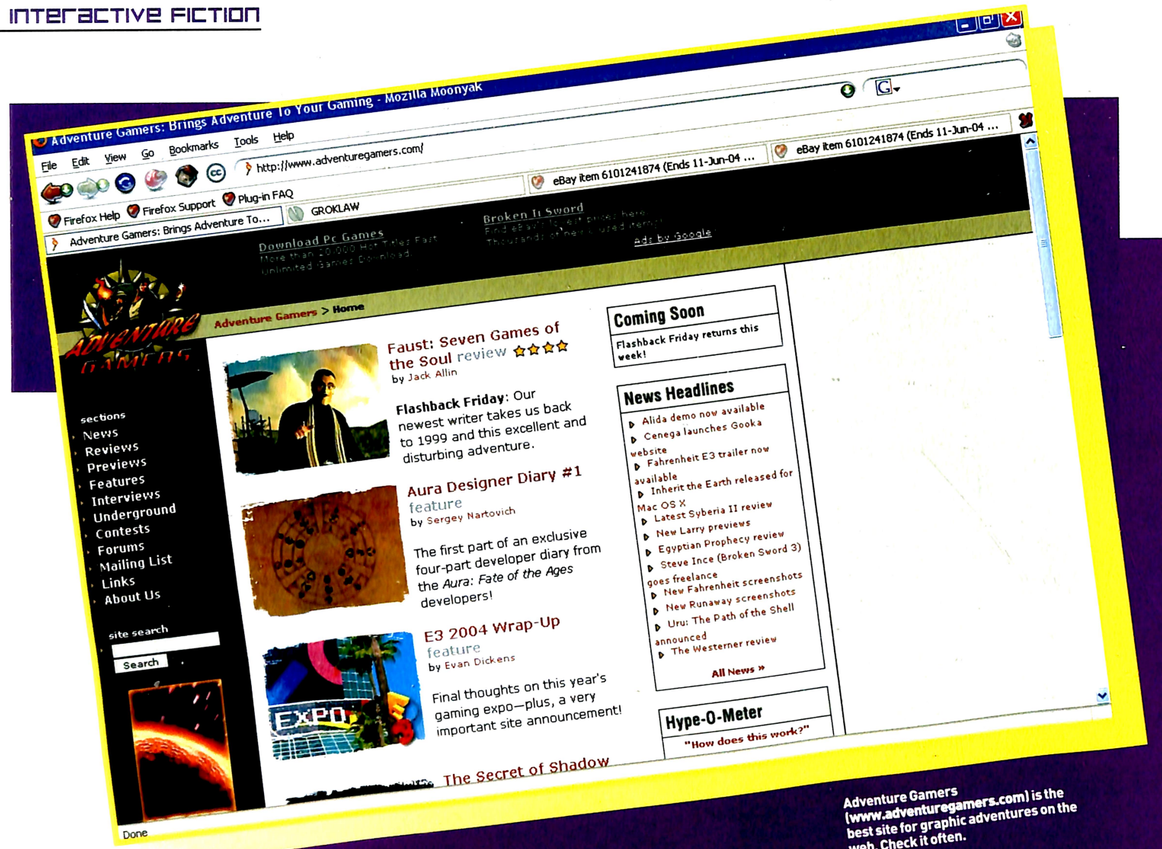
```
Object ornament "cat statue" Helipad_Landing
With name 'statue' 'cat' 'gnome' 'ceramic' 'creepy',
description "The creepy ceramic statue is in the shape of
a small black cat. Little flecks of light dance in its eyes.",
has scenery;
```

Here's where creating your objects starts getting difficult. The player will receive the above description when they type **examine statue**, or words to that effect. What if they now type **examine flecks of light**? We've mentioned them in the text, so we can't simply cough up a **You see no such thing** type answer. No, we have to keep working our way through the objects, filling in everything that the player might reasonably be expected to look at.

```
Object statue_eyes "eyes" Helipad_Landing
with name 'eye' 'eyes' 'flecks' 'pricks' 'reflections',
description "The sun reflects little pinpricks back out of
the cool ceramic eyes. Peering closer, you fancy you can
see little lenses inside. Of course, that couldn't possibly
be the case. There'd be alarms and panic and men with
guns were that the case.",
has scenery;
```

Simple rule of thumb – if you mention it, describe it. You don't have to go into any great detail, and indeed, your player won't want to

Zork Zero **– bringing graphics to the text adventure. After a fashion.**



Adventure Gamers (www.adventuregamers.com) is the best site for graphic adventures on the web. Check it often.

wade through an essay every time they open their eyes, but every little helps. Another thing you'll notice here is that we assume the player already knows they're referring to the statue. This is because we've ensured that this is the only object that will provide such keywords. Were we to sit the statue next to, say, a reflective pool of water, we'd quickly find ourselves in a confusing situation. Inform will compensate to some extent with responses like 'Do you mean the red door or the green door?', but ideally you just want the game to respond to the player's instruction without requiring any clarification.

// A-MAZING DEVELOPMENTS
The puzzle with a warning. Beware!

It seems like the easiest throw-in puzzle. You need an obstacle between the player and their destination, so why not add a nasty little maze for them to navigate? Here's three good reasons: they're annoying, frustrating, and as welcome as a flatulent skunk in a small lift. What makes it worse is that many authors have abused them in the past, creating multidimensional efforts where going north doesn't necessarily mean you return to the room you just walked south to reach, any breadcrumbs you drop onto the floor are magically disappeared as soon as your back is turned, and a bored player can't even use the Batman-favoured solution of just heading left at every intersection until reaching the exit. Mazes are bad. Mazes are evil. Avoid mazes like the plague.

CHARACTER MAPPING

Time to start work on some of the characters. These are typically referred to as NPCs – Non Player Characters. For now, let's worry about getting them up and breathing, then move onto the thorny issue of conversation. Here's our next block of code. You should find it familiar…

```
Object CitizenBlender "Citizen Blender"
with name 'Citizen' 'blender',
description "The blender glowers at you from
the table, its blades spinning as it looks you
over with its eyeless stare.",
has male animate;
```

Once again, we're dealing with an object. The only differences this time are **male**, which is self-evident, and **animate**, which sets up everything from how Citizen Blender should behave to what happens if the player tries to punch him. Of course, this is only a basic template; we can, and will rewrite great swathes of it by the time we're finished.

First, let's add a bit of life. The easiest way is to have Citizen Blender perform random actions between the player's moves. We do this using a command called **Switch**. In this instance, we feed it a random number between 0 and 40. If this comes out as 1, 2 or 3, we print out the relevant line. If not, nothing happens.

Note the use of symbols. In Inform, ~ represents speech marks, while ^ means **Go down a line**. We use these to prevent confusion with pieces of dedicated code.

```
each_turn
[; switch(random(40))
{1: "^The blender watches you carefully.";
2: "^ ~THE BLOOD OF THE WORKING CLASS
WILL DROWN THE BOURGEOISIE!~ shrieks
the blender suddenly. ~Later, there will be
cake.~";
3: "^The blender sits quietly, dreaming of
righteous vengeance and banana
milkshakes.";
}],
```

The next step up is to provide custom responses for the player's actions. By default, any object can be picked up. We don't necessarily want that to happen. Instead, we leap in before Inform's default actions have a chance to kick in. Here are two common verbs, both with our own unique responses. You can find a list of verbs in the Designer Manual (**www.inform-fiction.org**), and even create your own if necessary.

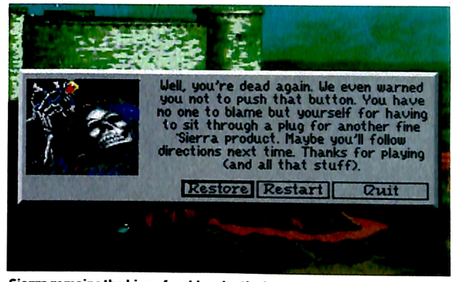
```
before
[; Take: "The blender whirls its rotor blades
at your questing fingers…";
Smell: "Smells like banana milkshake. Also
blood.";],
```

By combining these two methods, we have the perfect method for conversations. Because the Blender is alive – for our purposes, anyway – we use a

// ASHES TO ASHES, DUST TO RELOAD POINT.
When is it okay to kill off your player?

Death isn't always a bad thing. It encourages players to take care, adds a sense of danger to tense situations, gives you a chance to use your excellent joke about sticking your tongue into a light-socket, and of course, adds a sense of realism to your world. Just remember that you're not Hannibal Lector. The player should never, ever be put into a situation where they can't finish the game – for instance, eating a lump of cheese instead of using it to bait a mousetrap, or be incapable of escaping from a room because they forgot to pick up the rope back at the start of the game. Likewise, you should always give fair warning. They should never be killed simply for walking into the wrong room or flicking the wrong switch, unless it's obviously the wrong thing to do (such as casually wandering into the guards' barracks when escaping from a castle).

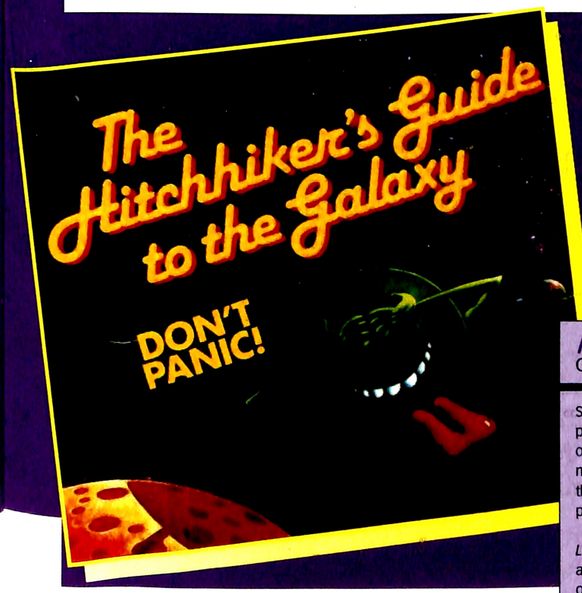
To actually kill the player off is easy. Simply add the line **deadflag = 3** after your detailed description of the dimwit's demise and Inform will handle the rest.


Well, you're dead again. We even warned you not to push that button. You have no one to blame but yourself for having to sit through a plug for another fine Sierra product. Maybe you'll follow directions next time. Thanks for playing (and all that stuff).
Restore
Restart
Quit

Sierra remains the king of sudden death. Just minding your business could be as dangerous as bungee jumping into a vat of hungry sharks.

BELOW
There's even books available on IF technique. You'll find them in a twisty maze of bookshelves, all alive.

BELOW LEFT
Bah! Everyone loves this game, and we've no idea why. It's not that great. Many would consider it 'rubbish'.



Life block to handle this. Simply fill in the words you want to ask about. If multiple entries lead to the same answer, separate them with a comma.

```
life [;
Ask: switch(second)
{
'blender': "~None of your business!~ snaps Citizen
Blender.";
'toaster', 'radio': "~Fools! I surround myself with foolish
fools!~";
```

Finally, we need to trap any non-existent conversation topics. We do this using a special case, called **default** – used if nothing else in the Switch statement fits the bill – before closing up the Life statement.

```
default: "~I have no interest in that,~ snaps Citizen Blender";
}],
```

With both rooms and characters in place, we can begin building up our stage. Next month we'll be making these environments more interactive, adding puzzles, and pulling the pieces together.

// RETURN OF THE KINGS
Games that came back, packing new graphical shine!

Sadly, few text adventures of old have ever managed to make a high-profile reappearance. The biggest success is the Zork series, kicking off with the (awful) full-motion-video adventure *Return to Zork*, moving onto the dark, moody *Myst* style *Zork: Nemesis* and finally the hilarious *Zork: Grand Inquisitor* – a real feast of twisted logic puzzles and adventure game parodies.

Most releases haven't been this successful. Sci-fi comedy *Leather Goddesses of Phobos* returned to the shelves in an apocalyptically bad graphical adventure, while fellow deep-space odyssey *Planetfall* was dispatched to the interstellar landfill long before launch.

By far the most successful conversion, believe it or not, was Sierra On-Line's *Softporn Adventure*. This fairly bland text adventure – notable only for inflicting a naked picture of Roberta Williams on the world – was picked up and re-written by Al Lowe as the first *Leisure Suit Larry* game. The series is soon to see its seventh installment, *Magna Cum Laude*.

Apparently a picture is worth a thousand words. Nobody said they had to be subtle ones.

Homeworld 2 The Basics

PART ONE

DIP YOUR TOE INTO THE MODDING OF THIS GREAT STRATEGY.

With its predecessor winning Game of the Year 1999, many wondered what Relic could possibly have in store as a follow up to the original *Homeworld*. *Homeworld 2* still retains those awe inspiring camera angles and soundtrack, but it's evident that the bar has really been raised with this title. Modification has been made much easier, thankfully, and we're here to show you how to squeeze every last drop out of potential modding capability. Over the coming few issues, everything will be dealt with from the tweaking of existing ships to the creation of a huge battle cruiser. To begin with though, let's start with a brand new map for *Homeworld 2*. You can generate them either randomly or custom them to excess. Read on and learn how!

When not arguing over which Chocobo noise is best [Kweh or wark?], Eve and Stuart spend their time tripping over their huge LAN cable, and playing too much Final Fantasy. Stu mods for the Homeworld community [aka Delphy] and Eve reviews for *PCF*.

DIFFICULTY EASY

TIME TAKEN 1 HOUR

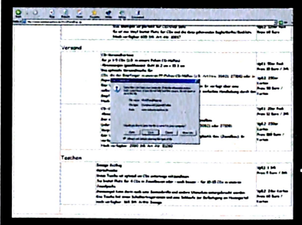
WHAT YOU NEED...
HW2 Random Map Generator
www.extreme-archive.de/HW2RandMap.zip

ON THE DISC

HW2: lush backgrounds, a stonking soundtrack and missions that'll have you tearing your hair out, while simultaneously making you wish it would never end.

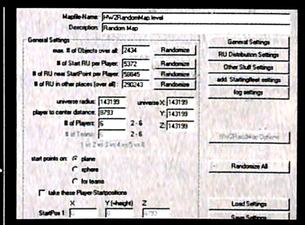
// INTRODUCING THE RANDOM MAP GENERATOR

Random maps = true test of gaming prowess. We show how to inject the pain!



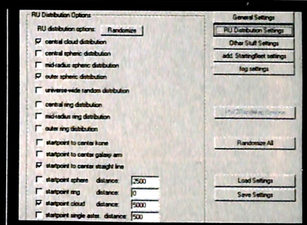
01 GETTING STARTED

Firstly, grab the ***Homeworld 2* Random Map Generator** from **www.extreme-archive.de/HW2RandMap.zip** It's the only program you'll need for this Part, but don't let that deter you – it's the key to unique and challenging maps without the science (that comes later in Part Two). Open it up and try not to feel swamped by its numerous options. Remember, this is supposed to be the easy part! Do the obvious by giving your map-to-be a name and enter a description.



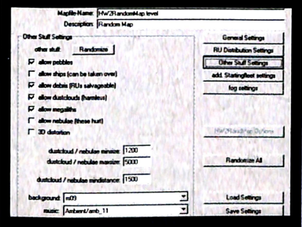
02 GENERAL SETTINGS

Time to decide on parameters. Click on **General Settings**, and here you can define map basics like number of players and resource units, player starting positions and map size. This is where you need to question the type of map you're creating. Will it be a cakewalk or a nightmare? Small with lots of players clashing for resources or will it be a large map with few players and thus take longer to complete? Remember these when defining the limits of your map.



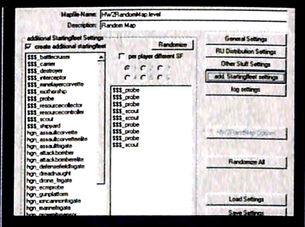
03 RESOURCE DISTRIBUTIONS

Resources are the backbone of any mission, so pay attention. Click on **Resource Distribution Settings**. Now you can tweak where and how asteroids and Resource Units (RUs) appear on your map and in what arrangement; for example, lump everything in the centre and you'll generate a map where all the players dive for the middle, constantly fighting over resources, whereas if you spread things out, players will in turn be further spaced apart and fighting will be sporadic.



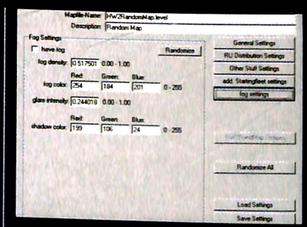
04 OTHER SETTINGS

Things are coming together now, so we can click on **Other Settings**. *Homeworld 2* is renowned for its opulent backgrounds so select something fitting for your own map from the dropdown list. While you're there, choose some nifty music too; a good map is a combination of all its elements. In this screen you can also turn on radioactive clouds: remember these from *Mission 11* at the Bentus Ruins? Also tweak the dust cloud and nebula details: these make good camouflage.



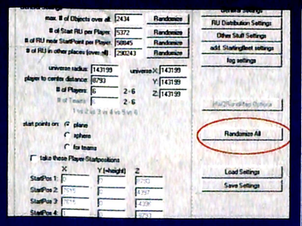
05 STARTING FLEET SETTINGS

This section allows even greater customisation to your map: choose exactly which type of ships you can give to each player. Confused? If you don't want everybody to start out with Fighters and have such monotony in the beginning, here you can delegate various classes of ships to each player. Handy when variety is the name of the game. Give a good mixture of everything from Corvettes to Interceptors to increase the challenge, or play nasty with a Battlecruiser!



06 FOG SETTINGS

More tweaking. Click into **Fog Settings** and know that, by fiddling with the options in here, Fog affects both visibility and ship sensors. It is thought that not knowing what lurks beyond the vicinity of your ships adds to the suspense – all that skulking in the shadows and seeing things just out of the corner of your eye. If a truly engaging map is what you're after, pile it on! Note: Glare is merely aesthetic and is graphic card dependant – use at your own discretion.



07 RANDOMISE ALL

Self explanatory; this option is only for the most advanced player. If selected, every option right across the board is randomised and you don't know what environment you'll be thrown into once the game commences. If all the tweaking so far have you chomping at the bit to play, then you can just load up the generator, hit **Randomise All**, plop it into your game and play – it's the fastest way to get a new map. But be careful.

08 SAVING YOUR MAP

To save your labour of love, click on **Create** and you'll be shown a general preview of your map. Don't click **Copy Level** to ***Homeworld 2* Directory**, as everybody who wants to play your map will need a copy of it. Instead, click **Cancel** and you'll find that a local copy of your map has been saved in the same directory as where you unzipped the map generator to. Copy this map and distribute unchanged to your fellow players.

09 CRUNCH TIME

It's almost time to see your work in action. Remember that map you just distributed with all the vigour of the ubiquitous AOL CD? Well, instruct your friends (you do the same) to copy it into **Homeworld2\Data\leveldata\multiplayer\deathmatch**. This is also the directory for all future maps you may create. Now load up *Homeworld 2* and bask in the glow of your random map making skills! See next month for more in-depth mapping.

OF THE NAME GAME

We explain the art of naming your map.

When you get in on the whole mapping scene, we bet you'll be churning out *Homeworld 2* maps faster than a Pentium EE. Before you know it, your directories will be full to the brim with myfirstmap.level, mysecondmap.level and mapwhereihaveabattlecruiser.level; there's got to be a better way for identifying your beloved maps, right?

Sure there is! Keeping things understandable and accessible when it comes to maps is key and it all starts with the very first step: the name. The major thing to remember is to include the number of players when naming, so you get maps with titles such as: 4p_MyMap.level or 3p_WorldOfPain.level, where "4" or "3" are the number of players allowed.

Feel all Martha Stewart at how tidy everything becomes (court case not included).

N SYNC

Like today's boy bands, every map must look exactly the same

In a nutshell, everybody wanting to play on a particular map absolutely must have the exact same file, whether it's created by random generation or made by hand. This is important and it's likely that the root of 90 per cent of map problems can be traced back to this simple fact. The only real bother is making sure everybody involved not only has the same file, but also has it placed in the correct directory and selects it on the map window. Simplistic stuff, but it is still surprising how often people think their maps don't work when it's really down to something so straightforward.

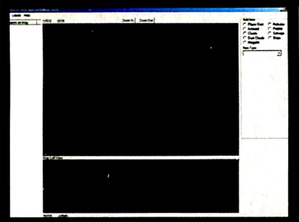
LABEL CONSCIOUS

If only it were this easy to be rid of branding.

When using Mission Boy SE for your custom map, you will notice that a lot of the yellow labels begin to overlap themselves as you add more and more objects. You'll be glad to know that there is a way to hide such labels. At the top of the screen, access the Labels menu and untick whichever set of particular objects you want to remain anonymous. There is a Hide All function too which could be worth checking out. Experiment to find your comfort zone. Remember, this option hides only the name, not the actual asteroid / cloud / nebula.

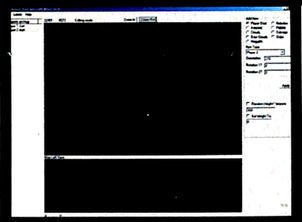
CUSTOM MAP CREATOR

Take your maps to the next level and get them just the way you want them!



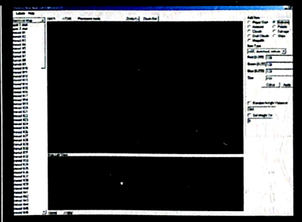
01 INTRODUCING

Open **MissionBoy SE** and find a new map waiting for you. The grid in the larger window is the top down view. The smaller window below is the front view. The grey panel on the left contains the objects you've already placed in the map (this is where you can delete objects too). The panel on the right contains the objects that can be placed. Clicking on **File** and **Options** lets you set map basics such as number of players, background music, map size etc.



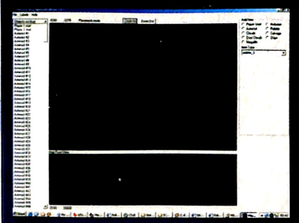
02 FIRST THINGS FIRST

The cluster of buttons at the top right are where you'll find all you'll need to compose your map. The first of these is **Player Start** which sets the exact position on the map that each player begins the game on. After clicking and plopping as many as you want into your map, you'll notice that you can change the direction players face in the right panel. Just enter the direction in degrees for each and that's it. In the example, our players face the centre.



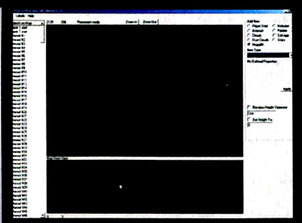
03 RESOURCES

Now we need to add what is probably the most essential component of a *Homeworld* map (besides having someone else to play against, but you'll have to wrangle that yourself): Resource Units (RUs). These are mainly found in asteroids, so click **Asteroid** and select which type you prefer to use. Asteroids 1 and 2 are not harvestable (no RUs), whereas 3-5 are. Click and drag onto the map to position: in our example we stuck them on the right hand side.



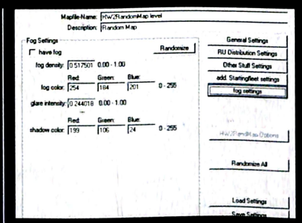
04 CAMOUFLAGE

The next things to add are the clouds and dust clouds. It's pretty much the same as before: select the respective button, select colour (purely aesthetic), then click and drag. You can add a bit of variety by checking the **Random Height box** so that, as you click and drag, your clouds are distributed at different heights. Remember, your map is in 3D which one can often forget if just watching the top down view. **The Random Height** also works for asteroids, pebbles, etc.



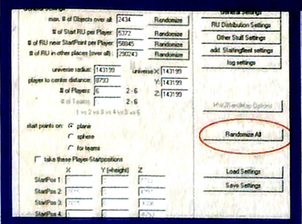
05 THE IMPRESSIVE BITS

Think Karos Graveyard from *Mission 8* and you think megaliths. These are enormous visual elements that can make a map much more realistic, e.g. we've added Saj Hulk Panels to the middle of our map. They're easy to place: just click **Megaliths** and select the ones you want. Remember, these pieces aren't harvestable but really add to a map. Couple megalith parts with some salvage chunks (step 8) and you can make a map appear to be the site of a ship explosion.



06 PUFFY BITS

Nebulas can affect the difficulty of a map, depending on their type and placement; however they do add another layer of spice and can even out the odds if your fleet leaves a lot to be desired. Click **Nebulas** and a dropdown appears. Pay attention because this can make or break a good game. If you choose a radioactive nebula (as featured in Mission 11), then all ships will take damage while passing through it. Choosing normal will give you RUs. Place at will.



07 PEBBLE BEACH

Visible in the **Sensor Manager** of a game as grey dots, pebbles give the impression of dust drifting in space. If placed correctly, pebbles can really enhance a map, but watch out. They eat processor and by placing too many of them may mean that only higher end PCs can run your map (XP3200s or Pentium4 3GHZ). With that in mind, click **Pebbles** and click drag to place. Note: writing rude words with them across the map is at the user's discretion, but highly advised.

08 SALVAGE SQUAD

The next objects that can be placed are **Salvage Chunks**. Click the relative button and the dropdown will ask you to choose between either large or small; this not only affects the shape of the Chunk (obviously) but it also affects the amount of RUs that the said Chunk will give you. The final button, **Ships Option**, lets you place additional crafts either for players or as a story plot i.e. give players gun turrets which can't be built straight away/fight over vacant ship.

09 EPILOGUE

That pretty much wraps it up for all the objects you can bung into a map, just remember that the more intelligent the challenge the better the gameplay (though nothing beats everybody with battle cruisers and just laying waste to each other – Stuart). Finally, to keep your pride and joy, save your map into this directory: **Homeworld2\ Data\leveldata\ multiplayer\deathmatch**. Drop by http://forums.relicnews.com to show off your latest creation.

MIDWAY

THE SUFFERING

FROM JULY 16 ON PC

PC CD

XBOX

PlayStation 2

ZOO Digital Publishing

THESUFFERING.MIDWAY.COM

The Suffering © 2004 Midway Home Entertainment Inc. Midway ... Home Entertainment ... ved. Distributed under license ... Midway Games Ltd. ... and ... of Sony Computer Entertainment Inc. Xbox and the Xbox ... or trademarks of Microsoft Corporation in the U.S. and/or ... are used ... licence from Microsoft. ... by ZOO Digital Publishing Limited. © 2004 ZOO Digital Publishing Limited. ZOO Digital Publishing and the ZOO Digital Publishing logo are trademarks of ZOO Digital Publishing Limited part of the ZOO Digital Group plc. All rights reserved.

Individuals are able to customise their appearance to a considerable level.

in hindsight

eGenesis a Tale in the desert

FROM THE VERY FIRST BRICK TO THE APEX OF THE PYRAMID, WE SPEAK TO THE PHARAOH'S CHIEF ARCHITECTS. BY KIERON GILLEN.

From the first second you logged onto *A Tale In the Desert*, it was clear that this wasn't just another massively-multiplayer game. For a start, it did the unimaginable in the videogame world and removed direct combat. Entirely. However, it wasn't the *Sims* Online's glorified chat-room. In this ancient Egypt, challenges awaited you. While initially it seemed to be about constructing in a grand co-operative venture – kind of a game of Settlers where you played one of the eponymous characters – players soon discovered that it was a far more political game than a world of simple, happy worker ants. Social puzzles abound, which had to be overcome, with personal gain faced off against group success. It's a game that challenged its player base in a way that no other game was even attempting. "*A Tale In The Desert* was the game I always wanted to play – and it's kind of ironic that it's the one game that I can't play", ruefully notes Andrew Tepper, President of the game's developer, eGenesis, "Obviously, that wouldn't be fair."

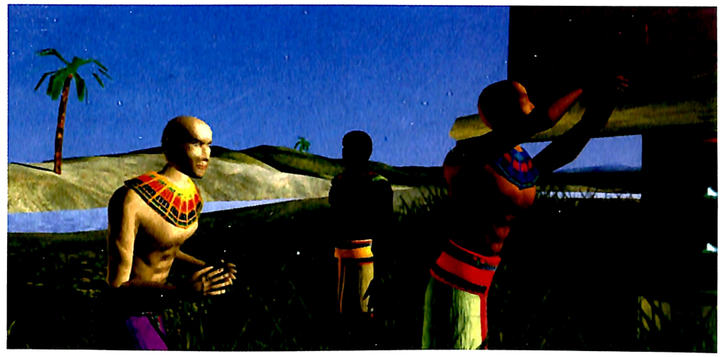
"When MMO games first came out there was the promise of all sorts of things that had never been tried before", he explains when asked about the game's origins. "The first games that came out were really just taking a single-player game and making them multiplayer. They didn't have any social puzzles. *Motor City Online* tried racing. There were lots of RPGs, but very few social puzzles, and I think that's the most interesting thing in real life." He references the American reality-TV show *Survivor's Interactive Dynamic* as an example of an exquisite social puzzle, and what he wanted to see in a game. "Nobody was doing it, and they're still not doing it in the sense that we're doing it," notes Andrew.

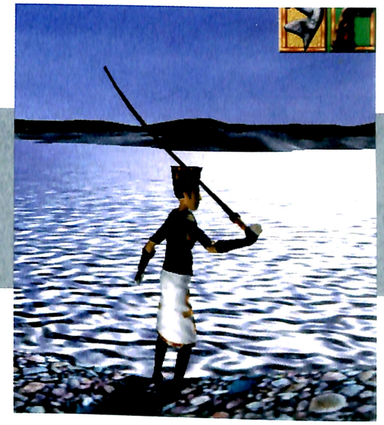
PUZZLE MUDDLE

An initial problem was actually devising these puzzles. "There are certain social puzzles that are well know – like the *Prisoners* dilemma – but what was commonly known at the time wasn't very large." One of *A Tale In The Desert*'s primary elements is the passing of tests, which pose a social problem for them to find a solution. An early example was the *Test of Marriage*. "A very simple puzzle: find one person you trust totally", explains Andrew. "The person who you can marry, who can use all your stuff – in fact, even log in as you. But in Egypt there's no concept of divorce, so it's the easiest test in the game to pass but it's the one that can totally screw up your game forever. And that's happened a few times. There's been several high profile marriages that really did". Similarly, the test of the Demi-Pharaoh. "The community has to elect one person they trust totally, who then has the power to ban – permanently exile – up to seven people at their whim, any time," notes Andrew.

Abstractly these are simple challenges, but its implications could lead to all manner of human problems. To see how players reacted to this, consider the Test of the Bureaucracy. "A few people are assigned the precious opportunity to start a bureaucracy," explains Andrew. "You do well in this test by growing it as big as you can. It's an organisational puzzle. Anyone in the bureaucracy can take it over, and so gain its resources. It's easy for them to do it. In fact, people who are lower ranked in the bureaucracy do it easily. They're gaining lots of points for you, but getting few for themselves. And there's also more of them, with lots to gain by taking over, ruining your hard work. It's a test to see how well you manage a large organisation."

"At the start they simply said 'Most people in Egypt are trustworthy', and threw it open," Andrew notes. "They recruited friends. They told their friends to recruit friends. Let's make it as big as you can. Of the first six, five were taken over by either Griefers (*trouble causing players* – Ed) or new players who didn't realise what they were doing." Obviously, time for a rethink. "The second reaction was 'Oooh it's a hard test. All the bureaucracies get taken over'. They started thinking of having a structure. Everyone has a different approach, but one is if you're considering recruiting a new player into the bureaucracy the player has to have been in Egypt for at least two months, passed at least one test and then obtained approval from the other two people at your own level. Each evolves its own rules, by the players, trying to prevent the organisation from growing out of control".





LEFT Male Pattern Baldness abounds on the banks of the Nile.

BELOW LEFT Set up a printing press and publish your own fanzine. Really.

BELOW After a hard day paraohing, relax with a fishing trip.

THE SECOND TELLING

Learning your limitations.

***A Tale In The Desert*, somewhat uniquely, was a MMO game with an ending always built into the structure. This lead to the second telling, for which eGenesis took time to fix some of the problems with the game. "Number one thing was travelling," Andrew notes, "It's much too time consuming in *Tale: 1*. It's a big map, and you have to get to places on foot, and that tended to chew up a lot of your play time."**

"Another thing we miscalculated, because of the technical limitations of the server, was not being able to put thousands of buildings on the screen," Andrew continues, "We won't be able to render them. We can't push that much data to a client. We used crowding where you can only build to a certain density, which pushed people apart: on the screen in *Tale 1*, while there may be hundreds of players online, you only see a few. We've made a low-level technical improvement to the server, to give things a city feel. Previously you had a camp that was a loose collection of buildings. Now you build a work-shop – but it's expandable. You decide what shape you build the workshop itself and then you build the components of the skills you need."

While the social puzzles were at the heart of the design, it took a while for people to realise this, preferring to turn their attention to the more obvious. "It was a foreign concept to people at first," remembers Andrew. "What they latched onto at first was building. It's something everyone was familiar with, and we have hundreds of things to build and craft. Players learned to do this very quickly, and enjoyed it. At first it seemed the game was just about building." Which, of course, it wasn't. Not purely, anyway. "If I had to honestly note down the failures of the game, teaching people to get involved in the social side of the game was a little harder than I expected," comments Andrew. "There's so much to build that people burned out, and many people tried to do very large projects either by themselves or in a small group of other players. They didn't try to co-operate region wide to get 100 people working on a project."

AUDIENCE APPEAL

The uniqueness of *ATITD* makes it hard for people to grasp. How do you explain to new players, concepts which aren't easily categorised. "I still don't know the best way to reach people," explains Andrew. "That's the biggest problem we have. The game appeals to people who like building. We have a lot of women playing the game – 27 per cent by accounts, 40 per cent by hours played. Reaching female gamers? It's a difficult group. We have academics and professionals, but I don't know who our core audience are."

But there's much to be pleased with. "The way the community has formed is everything I hoped it would be," Andrew smiles. "When a new player comes into Egypt they typically get a very warm reception. You don't typically encounter griefers at an early stage. The fact is, in order to make progress in the game, you have to co-operate with other people and make them want to help you: that leads to creating a culture that does welcome new people. I see over and over again people who try the game, and maybe go and try another game, but come back. They always come back to *A Tale In The Desert*. I hear that there's something about the community's friendliness that isn't there in most other games. I expected it in *A Tale in the Desert* and it worked out exactly as I hoped."

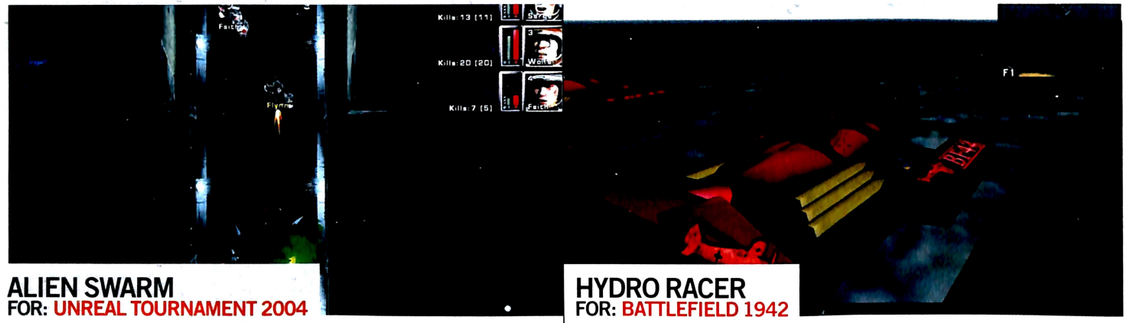
One of the more interesting aspects of the game was the legal system, where the players voted on their own laws, which then would be enforced in the game; leading to some surprising results for the developers. "In real life, I'm a Libertarian," claims Andrew. "I prefer very little government – a very basic capitalism is my own philosophy, and people should be free to do whatever they want, even if you're going to hurt yourself by doing it, and the Government doesn't tell you what to do. What I expected in *A Tale In the Desert*, well, in real life, Governments tend to start off small and grow and grow and grow. I expected the same thing to happen in the game world. I expected Griefers to show up, people pass laws to deal with them, but eventually get a taste for passing laws. The Government would get out of control, and be relatively oppressive. And that didn't happen at all. They continue, even now, to be very wary about the kind of laws they pass. You don't see that in real life. As a Libertarian, I'm pleasantly pleased. Though perhaps it's not as interesting as if they did really screw things up for themselves…"

What advice should be given to a game designer thinking of making a similarly co-operative game? Well, look a little closer and realise that it's not really just about that. "The intent was never to make a purely co-operative game," Andrew claims. "We give game projects that people have to co-operate in, and then we give them reasons why they would want to act selfishly. The tension between those two, knowing that you have to co-operate to get certain things done, but also knowing that co-operating isn't the personal best route to do things, creates an interesting decision. Don't try and create a purely co-operative game. Give reasons to co-operate and reasons to not, and make each one of those as meaningful as the other. Co-operating doesn't just mean creating a monster so big that you need fifty players to defeat. You have to be more creative than that. Also have rewards for individual accomplishment, so that there is a tension. Interesting things will emerge."

"Actually, that's a more general approach," he continues. "If you set rules up where, you as the designer, don't know the optimal way, very often it creates some pretty interesting behaviour. Not always however. Sometimes the players are, as a whole, smarter than you are and there is an optimal solution which you have just missed. But very often you get all kinds of cool and innovative stuff that no-one has ever thought of, and then all of a sudden, they think you're an absolute genius."

odd mods

MARK HOLTHAM INVESTIGATES THE OFF-THE-WALL WORLD OF WEIRD AND WACKY MODS.



ALIEN SWARM
FOR: UNREAL TOURNAMENT 2004

If *Gauntlet*, the classic top down view game of last century, was to get an update that was truly in honour of its playability and added cooperative multiplayer mischief, then *Alien Swarm* would be that game. This class-based total conversion of *Unreal Tournament 2004* is one of the most complete mod releases to be seen for quite some time. The game hosts a single-player campaign and cooperative multiplayer mode through either a single level or the entire campaign. While the single-player does offer a challenging experience, it's the multiplayer aspect of the game that really shines. Playing with friends and shouting in horror as you hose down waves of bugs like aliens is extremely enthralling. If you own *UT2004*, this is a mod not to miss out on. www.blackcatgames.com/swarm

HYDRO RACER
FOR: BATTLEFIELD 1942

Hydrofoil racing is a relatively unknown sport on the shores of old blighty. The thought of having the squirrel's choice delicacy retract into one's abdomen while spectating from the UK's chill blasted coast can bring instant displeasure. This mod does try to bring a more enjoyable experience than the average retail flop based on this sport. Players bounding around *Battlefield 1942*'s watery maps using islands as waypoints is a standard racing affair, but the addition of missiles, torpedoes and machine guns is inspired. Not quite keeping to the purity of the sport, but it raises the level of entertainment and competition. If you're a watersports fan or just interested in something different, then give this a go, even if it is a tad wet behind the ears. www.planetbattlefield.com/bf2029/hydroracers

MATRIX MOD
FOR: QUAKE III

As a movie, *The Matrix* was revolutionary, but as a game it never lived up to our over-active expectations. Since the Wachowski brother's first slowmo-tastic movie, nearly every moddable first person shooter had the honour of gaining a *Matrix* adaptation. This mod is for *Quake III Arena* and it happily retains *Quake*'s traditional fast and spontaneous combat. The core of the game is the Bullet Time feature: something we have yet to decide is a help or a hindrance. Slow motion comes without reason and it acts with a complete random nature: is it a bug or the intended purpose, we wonder? In the end it doesn't even matter, as when you are engaged with friends, simply sit back and enjoy the bullet effects as you re-enact the lobby scene.
www.planetquake.com/matrixq3

CODE RED
FOR: QUAKE III/FREEWARE

When downloading a new mod, your mind wonders what lies in wait. *CodeRED* creates a warped mental image of an interactive Tim Burton film. It's a fairly generic adventure but the enemies seem to have been lifted directly from *Mars Attacks*. As in the film, the sight of the martians mincing around the cramped levels provides more than a hint of humour. Yet the most interesting item within the game is the crosshair: it's HUGE. While the obstruction can be adjusted in the settings, you just don't get that satisfying feeling of wanting to club each and every alien with the Death Star sized weapon H.U.D, as you would if you used one of the smaller selection of crosshairs.

A copy of *Quake III* is not required, so for a quick single-player fix, try this… www.planetquake.com/cor/codered

GAMEMAKER NEXT MONTH »

HOMEWORLD 2, BATTLEFIELD MODELING, CALL OF DUTY ONLY WITH *PCFORMAT* ISSUE 165 – ON SALE 12-08-04

"PlayStation" are registered trademarks of Sony Computer Entertainment Inc. © SOCOM II (c) 2003 Sony Computer Entertainment America Inc. Developed by Zipper Interactive. The U.S. Navy provided technical assistance, but does not officially endorse this product.

PlayStation 2

SOCOM II

Prepare to do battle with the rest of Britain in SOCOM II. Get your team together and join forces online. Work as a unit to survive. And remember: either we all go home, or no-one goes home.

fun, anyone?
PlayStation 2
www.socom2-game.com

5" OF DVD PLEASURE IN THE PALM OF YOUR HAND

YAMADA PDV 500 PORTABLE MULTI-REGION DVD PLAYER - PERFECT FOR SUMMER HOLIDAYS

- MULTI-REGION DVD PLAYER
- 5" ACTIVE MATRIX TFT SCREEN
- DOLBY 5.1 & TV OUTPUT
- ULTRA SLIM REMOTE CONTROL
- AUTO / PAL / NTSC SYSTEM
- IN-CAR KIT INCLUDED
- RECHARGEABLE Ni-MH BATTERY

YAMADA PDV 500 PORTABLE DVD PLAYER

£126.99

INCLUDES VAT - SAVE £40 OFF RRP

ANY PC MOVIE FILE ON YOUR TV IN 5.1!

DIVX, AVI, MPEG, XVID + INTERNET FILES

YAMADA DVX 6600 WILL PLAY VIRTUALLY ANY DISC IN ANY FORMAT, INCLUDING INTERNET FILES!

PLAY REGULAR DVD MOVIES (ANY REGION & RCE) & PC MOVIE FILES IN DOLBY 5.1 ON CD-R(W), DVD-R(W), VCD, S-VCD & MORE. PLAYS AUDIO, JPEG, MP3 & WMA CDS TOO!

YAMADA DVX 6600 MULTI-REGION DVD PLAYER

£69.99

INCLUDES VAT - SAVE £40 OFF RRP

UKDVD.COM

0870 725 9999

Special Reserve
Discount Network
Specialreserve.com

MAIL ORDER SALES - CALL:
0870 725 9999

SHOP WITH US ONLINE AT:
UKGAMES.COM

OR VISIT ONE OUF OUR AMAZING
8 BIG DISCOUNT SHOPS
OPEN UNTIL LATE

BASILDON
Festival Leisure Park, Basildon, Essex
☎ 01268 287 776

BRISTOL
351 Gloucester Road (A38), Bristol
☎ 0117 924 5000

CHELMSFORD
43 Broomfield Road, Chelmsford
☎ 01245 348 777

EGHAM
168 High Street, Egham, Surrey
☎ 01784 473 444

NORWICH
10/14 Plumstead Road East, Norwich
☎ 01603 300 004

NOTTINGHAM
164 Derby Road, Stapleford, Nottingham
☎ 0115 949 1000

SAWBRIDGEWORTH
The Maltings, Station Road, Sawbridgeworth, Herts
☎ 01279 322 399

UPMINSTER
Just 6mins from J29, M25. 209/213 St Mary's Lane, Upminster, Essex
☎ 01708 22 55 44

>> STOP PRESS <<

McAFEE VIRUS SCAN 7 (2004 EDITION)

£7.99

SAVE £22

AVAILABLE AT

UKPC.COM

0870 725 9999

>> STOP PRESS <<